Oracle FLEXCUBE Direct Banking

System Handbook – Volume III – Channel Layer
Release 12.0.2.0.0

Part No. E50108-01

September 2013

ORACLE®

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax:+91 22 6718 3001
www.oracle.com/financialservices/

Contents

1. Preface 1
   1.1. Intended Audience 1
   1.2. Documentation Accessibility 1
   1.3. Access to OFSS Support 1
   1.4. Structure 1
   1.5. Related Information Sources 2
2. About This Document 3
   2.1. Glossary Of Terms 3
      2.1.1. LICENSEE 3
      2.1.2. IMPLEMENTER 3
   2.2. TERMINOLOGY 3
   2.3. Abbreviations 4
   2.4. Conventions 5
3. Channel Management Tier 6
   3.1. Session Management 6
   3.2. Audit Trail 7
4. Channel Management Tier – Extensions and Configurations 8
   4.1. Configurations 8
   4.2. Extension Data Security 9
   4.3. Extension AUDIT TRAIL 10
   4.4. Extension Channel plugin 11
   4.5. Extension Service orchestration 12
   4.6. Wizard configuration 13
      4.6.1. Database Configurations 13
      4.6.2. Names, Descriptions and Skip text 14
      4.6.3. Limitations 15
   4.7. Wizard configuration 15
      4.7.1. Database Configurations 16
      4.7.2. Names, Descriptions and Skip text 17

4.7.3. Limitations 18
4.8. Extension Audit Logging 18
4.9. EXTENTION CUSTOM Validations 19
5. Interfacing with External Systems 20
5.1. Integration with Live Help by Oracle ATG 20
5.1.1. Live Help Account Setup 20
5.1.2. Live Help Availability Configurations 21
5.1.3. Transaction Level Configurations 22
5.1.4. Error Level Configurations 23

1. Preface

1.1. Intended Audience

This System Handbook (Volume III – Channel Layer) is intended for the following audience:

- Application Architects
- End to End Designers
- Business Service Detailed Designers and Developers
- Implementation Partners

1.2. Documentation Accessibility

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

1.3. Access to OFSS Support

https://flexsupp.oracle.com/

1.4. Structure

This document, termed Oracle FLEXCUBE Direct Banking System Handbook, is a single reference for the product information which can be managed, configured, extended, by external parties, to implement, customize or rollout the product to a financial institution.

This is not an Implementation Guide but a System Handbook to explain low level details of how certain key features are implemented within the solution and how these could be extended, customized as appropriate to meet the requirements of the implementation.

This document is intended to provide a set of principles, guidelines and parameters for configuration and extending Oracle FLEXCUBE Direct Banking to meet the . As such, this document does not go into detail regarding the context and background of a number of design decisions but explains the extensibility features and provides insight into the design guidelines and principles for external parties to leverage and develop the required extensions in a non invasive way to the primary features and functionality of the application.

This document is segregated into five Volumes

1	Volume I – Core and Architecture
2	Volume II – Presentation Layer
3	Volume III – Channel Layer
4	Volume IV – Business Service Layer
5	Volume V – Host Interfacing Layer
6	Volume VI – Origination and Peer-to-Peer Payments

1.5. Related Information Sources

For more information on Oracle FLEXCUBE Direct Banking Release 12.0.2.0.0, refer to the following documents:

- Oracle FLEXCUBE Direct Banking System Handbook – Volume I
- Oracle FLEXCUBE Direct Banking System Handbook – Volume II
- Oracle FLEXCUBE Direct Banking System Handbook – Volume IV
- Oracle FLEXCUBE Direct Banking System Handbook – Volume V
- Oracle FLEXCUBE Direct Banking System Handbook – Volume VI

2. About This Document

2.1. Glossary Of Terms

The following terms are some of the key terms used within the document for identifying the actor for the various actions mentioned within this document.

2.1.1. LICENSEE

The LICENSEE is the Financial Institution, Application Services Provider or the Bank which has licensed the Oracle FLEXCUBE Direct Banking application and shall rollout the solution to its customers as an internet and / or mobile banking channel.

2.1.2. IMPLEMENTER

The IMPLEMENTER is the Implementation Partner, Vendor, Application Service Provider or the LICENSEE themselves who is responsible for rolling out, configuring, extending or developing on Oracle FLEXCUBE Direct Banking.

2.2. TERMINOLOGY

The following terms and terminology is used within the documents to explain underlying processes, components, actions, actors etc.

Term	Definition
Business Service	A Business Service or a Transaction Service is a coarse-grained component that delivers a particular service contract. The Service Interfaces and that make up the contract are each implemented by their particular Service Endpoints.
POJO	A Plain Old Java Object (POJO) is exactly what it says. The term is used to differentiate these simple objects from more specific or complex types such as EJB classes. For example, when creating an EJB, a specific class must implement the SessionBean interface. However, that class will often delegate much of its functionality to one or more POJOs to aid maintainability and reuse of functionality.
Service Implementation or Service Endpoint	A Service Implementation is a concrete implementation of a Service Interface.
Service Interface	A Service Interface is a cohesive set of Service Methods that are grouped together in the anticipation that they will be commonly used together by a

	consumer. For example, the Service Interface for the FundsTransferService would contain a set of Service Methods that perform different types of immediate money transfer between two accounts.
Service Method	A Service Method takes the form of a Java method implemented by the Service Implementation and the Service Delegate. The consumer of the service will invoke one or more Service Methods to help perform part of a business process.
Extension Schema	The **Extension Schema** is a term used for the separate database schema as deployed by Oracle FLEXCUBE Direct Banking to allow IMPLEMENTERS to extend the Oracle FLEXCUBE Direct Banking application as per their needs.

2.3. Abbreviations

Acronyms	Description
FCDB / FC DB / FC Direct Banking / Direct Banking	Oracle FLEXCUBE Direct Banking
Java EE / JEE	Java Enterprise Edition
Java SE / JSE	Java Standard Edition
Java ME / JME	Java Mobile Edition
DBA	Database Administrator
XML	Extensible Markup Language
XSL	XML Stylesheets
TCP	Transmission Control Protocol
HTTP	Hypertext Transmission Protocol
HTTPS	Secured Hypertext Transmission Protocol
SSL	Secured Socket Layer
IDS	Intrusion Detection System

2.4. Conventions

- The diagrams and / or text in this document may contain colour to communicate or highlight additional information. However, the content of this document is retained when rendered without colour. Specific references to colour can be ignored if necessary.

- The technical terminology relating to the Oracle FLEXCUBE Direct Banking solution is aligned as much as possible to standard definitions or should be defined in the Glossary of Terms. Any deviations from standard terminology are either noted in the Terminology Section, or in context of usage.

- Some sections may contain additional notes and caveats included with the body text. For general and contextual information, these notes are contained within document footnotes. Any notes that have important implications or detailed recommendations are denoted by the information symbol (ⓘ). Important caveats are denoted with the warning symbol (💣).

- Some sections may contain examples included with the body text. Such examples are denoted by the use of shading and the introductory word “EXAMPLE”.

3. Channel Management Tier

The Channel Management tier performs the Session Management activities, and controls the lifecycle of any channel session. The Tier has access to the Local Database (LDB) used within the Oracle FLEXCUBE Direct Banking solution and uses the same for session related activities.

Audit Trail logging and Data Level Security are features of this layer before the request it passed to the Business Services for transaction processing.

The Channel Management Tier communicates with the Business Tier using Web Services by default. The options are available in Oracle FLEXCUBE Direct Banking to configure the Channel Management Tier to point to Service Orchestration Engine like FLEXCUBE Connect or a BPEL Engine to choreograph the transaction processing using simple services developed in Oracle FLEXCUBE Direct Banking or external systems.

The default deployment option does not require FLEXCUBE Connect or any BPEL Engine for the solution to be implemented.

3.1. Session Management

The USERSESSION table stores the primary session of the user. This table stores session information which is retrieved by Channel tier before accessing Business tier.

DATA SECURITY

Channel tier Framework provides developer capability to use predefined processing features for data storage which can be used to provide simple data-security during a transactional workflow.

Application allows configuration capability to store a request & read the request back before proceeding with business processing. This serves as a data-security feature as follows:

1) Typically a transaction workflow consists of data-entry screen, verification screen and confirmation screen.
2) On data-entry screen, user keys-in intended data & submits the same.
3) The submitted data is persisted by channel tier. This is based on configuration.
4) The data is validated by business tier.
5) If validated, verification screen is displayed showing data as submitted by user. User needs to verify the data & choose to confirm/change it.

6) If user chooses to confirm, no user data is submitted from verification screen. The user request data is rather picked from the persisted storage & forwarded for processing.
7) This feature ensures user that data cannot be tampered on confirmation.

3.2. Audit Trail

Channel tier features highly useful & configurable auditing capability. Both request into the channel tier & response by channel tier can be audited. The auditing can be controlled at workflow level. The auditing component can be extended to enrich the auditing behavior. As out-of-box, channel tier auditing component allows auditing to table `auditlog` in local database schema. Please refer to extension section for detail.

4. Channel Management Tier – Extensions and Configurations

4.1. Configurations

Channel tier uses XML based property file `fcat-config.xml`. The property file typically contains information to locate local database schema, service orchestration information. Since this file contains database information, this file should be encrypted using tool provided for same. Please refer Installation documents for same.

Sample property `file fcat-config.xml`

```
<FAML>


<FCON.A2.CONNPOOLED>N</FCON.A2.CONNPOOLED>

<FCON.A2.JNDI.NAME>A1</FCON.A2.JNDI.NAME>

<FCON.A2.LDB.DRIVER>oracle.jdbc.driver.OracleDriver</FCON.A2.LDB.DRIVER>

<FCON.A2.LDB.URL>jdbc:oracle:thin:@dbhost:1521:sr1</FCON.A2.LDB.URL>

<FCON.A2.DATABASE.NAME>ORACLE</FCON.A2.DATABASE.NAME>


<FCAT.CONNECT.INVOCATION.MODE><![CDATA[E]]></FCAT.CONNECT.INVOCATION.MODE>

<FCAT.CONNECT.JNDI><![CDATA[ServiceEndPointEJB]]></FCAT.CONNECT.JNDI>

<FCAT.CONNECT.EJB.java.naming.factory.initial>weblogic.jndi.WLInitialContextFactory</F
CAT.CONNECT.EJB.java.naming.factory.initial>

<FCAT.CONNECT.EJB.java.naming.provider.url>t3s://servhostname:7003</FCAT.CONNECT.EJB.j
ava.naming.provider.url>

<FCAT.ORCH.PLUGIN.T>com.iflex.fcat.channels.plugins.TransactionBeanPlugin

</FCAT.ORCH.PLUGIN.T>

<FCAT.ORCH.PLUGIN.C>com.iflex.fcat.channels.plugins.FLEXCUBEConnectHostPlugin

</FCAT.ORCH.PLUGIN.C>

<FCAT.ORCH.PLUGIN.X>com.iflex.fcat.channels.plugins.FLEXCUBEConnectHostPlugin
```

```
</FCAT.ORCH.PLUGIN.X>

</FAML>
```

💣 Changing any property in fcat-config.xml will require server restart for change to take effect.

💣 Since this file contains database information, this file should be encrypted using tool provided for same. Please refer Installation documents for same.

4.2. Extension Data Security

Channel tier Framework provides developer capability to use predefined processing features for data storage which can be used to provide simple data-security during a transactional workflow.

The above configuration resides in table `MSTCHANNELATS` which holds configuration for a transaction's workflow.

Column `MSTCHANNELATS.FLAGPREPROCESS`: Enumerations mentioned below. The processing is applied on the request path before generating the service request XML.

Pre Process	Description
1	Fetches only request data stored in table `usersessiondata` and add it to current request data. The request data to be fetched from `usersessiondata` is identified using request parameters "`fldSectionId`" & "`fldDataId`"
2	Persist the current request data in `usersessiondata` & return `fldDataId` (identifier to the data) in response.
5	Fetches only response data stored in table `usersessiondata` and add it to current request data. The request data to be fetched from `usersessiondata` is identified using request parameters "`fldSectionId`" & "`fldDataId`"
6	Fetches Both request and response data stored in table `usersessiondata` and add it to current request data. The request data to be fetched from `usersessiondata` is identified using request parameters "`fldSectionId`" & "`fldDataId`"

Column `MSTCHANNELATS.FLAGPOSTPROCESS`: The processing is applied on the response path before response is passed to Presentation tier. Enumerations mentioned below.

Post Process	Description
1	Fetch request and response data stored in table `usersessiondata` and add it to current response. The request and response data to be fetched from usersessiondata is identified using request parameters "`fldSectionId`" & "`fldDataId`"
2	Persists the current request and response data in `usersessiondata` & return `fldDataId` (identifier to the data) in response.
4.	Updates entire SectionID Node with Response(Does not retain request if it exists in usersessiondata) and add it to current response. The Response data to be updated in usersessiondata is identified using request parameters "`fldSectionId`" & "`fldDataId`".
5.	Fetches only response data stored in table `usersessiondata` and add it to current response. The response data to be fetched from usersessiondata is identified using request parameters "`fldSectionId`" & "`fldDataId`"
6.	Updates only Response(Retains request if exists) data in `usersessiondata` and selects updated response and previous request(if exists) to current response being send to Presentation tier. The Response data to be updated in usersessiondata is identified using request parameters "`fldSectionId`" & "`fldDataId`".

4.3. Extension AUDIT TRAIL

Channel tier features highly useful & configurable auditing capability. Both request into the channel tier & response by channel tier can be audited. The auditing can be controlled at workflow level. The auditing component can be extended to enrich the auditing behavior. As out-of-box, channel tier auditing component allows auditing to table `AUDITLOG` in local database schema.

Auditing component implements interface "`com.iflex.fcat.channels.audtrl.AuditTrailHandler`". This allows extension provides capability to extend & develop new auditing behavior.

The audit handler to be used by Channel tier for auditing is configured in XML property file `fcat-config.xml`

```
package com.iflex.fcat.channels.audtrl;


public interface AuditTrailHandler {


public void handleAuditLog (
```

```
  Connection  p_con

, AuditData  p_audit_data

) throws Exception;


}
```

Auditing can be configured at workflow level in column MSTCHANNELATS.AUDITREQUIRED .

“Y”: Auditing required

“N”: No Auditing.

4.4. Extension Channel plugin

The channel tier plugin defines the contract between the channel tier and the service tier.

Various channel plugins are available in the framework which can be used according to the requirement. Below properties are configured in channel tier property file i.e. fcat.config to define the mapping between plugin type and the class being used for it.

```
<FCAT.ORCH.PLUGIN.T>com.iflex.fcat.channels.plugins.TransactionBeanPlugin

</FCAT.ORCH.PLUGIN.T>

<FCAT.ORCH.PLUGIN.C>com.iflex.fcat.channels.plugins.FLEXCUBEConnectHostPlugin

</FCAT.ORCH.PLUGIN.C>

<FCAT.ORCH.PLUGIN.X>com.iflex.fcat.channels.plugins.FLEXCUBEConnectHostPlugin

</FCAT.ORCH.PLUGIN.X>
```

Channel plugin types are three, namingly T, C and X. It is defined in the database in table "mstchannelats", in the column "typPlugin".

PLUGIN TYPES

Plugin Type	Class Name	Description
T	TransactionBeanPlugin	This class provides the plugin for interfacing the Oracle Flexcube Direct Banking Transaction Bean, an EJB based common interface for invoking the transactional business component.
C	FLEXCUBEConnectHostPlugin	This class provides the plugin for interfacing the Oracle Flexcube Direct Banking Channel Management tier with the Oracle Flexcube Direct Banking Services tier for all business transactions. The class provides the ability to make Web Service calls to the required.
X	XXX class	The class being used is 'FLEXCUBEConnectHostPlugin' for this plugin type. This provides the facility to use customized plugin environment.

4.5. Extension Service orchestration

The Service Orchestration Tier is an optional and logical tier within the Oracle FLEXCUBE Direct Banking architecture which can be introduced if multiple Business Services are required to be managed using external orchestration module.

Following configurations allow orchestration management:

Column MSTCHANNELATS.FLGORCH

Value	Description
'C'	This denotes the no external orchestration is needed & the flow should be delegated to business tier for further processing.
'B'	This denotes that external orchestration BPEL is needed. Further properties to locate the orchestration provider should be configured in fcat-config.xml file. Please refer document "*Oracle_FLEXCUBE_Direct_Banking_Parameter_Sheet*" for details on properties to be configured.

4.6. Wizard configuration

Oracle FLEXCUBE Direct Banking provides a functionality wherein specific Wizards can be configured. This is an optional configuration to specify the workflow for any given set of transactions.

The following configurations are needed to enable wizards:

4.6.1. Database Configurations

MSTWIZARD

This table holds the base/master configuration for Wizards. The configurations in this table determine the condition for which the Wizard is to be invoked. Following are the columns and the expected values:

Column Name	Description	Expected Values
ID_ENTITY	The Entity ID for which the wizard is to be invoked	F001, B001, B002…. (Must be a correct entity configured in MSTENTITYUSERCHANNELS table)
USERTYPE	The User Type for which the wizard is to be invoked	ECU, EN1… (Must be a proper user type as configured in MSTENTITYUSERCHANNELS table)
IDCHANNEL	The Channel/Device ID for which the wizard is to be invoked	01, 42, 43… (Must be a proper channelis as configured in MSTENTITYUSERCHANNELS table)
IDREQUEST	The Request ID for which the wizard is to be invoked	RRLGN01 (Must be a proper idrequest as maintained in MSTCHANNELATS table)
IDWIZARD	The unique identifier for the wizard. This is referenced in WIZARDDETAILS table.	FLOGIN, FLLOGINRET (Any character string allowed)
START_CONDITION	The fully qualified java class name that defines the condition under which the wizard can be invoked.	This must be the fully qualified name of a Java class that implements the com.iflex.fcat.channels.wizard. WizardVerifier interface. This column may be left *null*. If null no condition would be checked other than matching the ID_ENTITY, USERTYPE, IDCHANNEL and IDREQUEST for the incoming request.

WIZARDDETAILS

Column Name	Description	Expected Values
IDWIZARD	The unique identifier for the wizard.	FLOGIN, FLLOGINRET (Any character string allowed. The same needs to be referenced in MSTWIZARD.IDWIZARD column)

STEPSEQ	The step sequence number of the particular step in the wizard.	0, 1,… (Must be a non-negative integer. The first step must always be 0. All steppseq for a given wizard must be sequential in the order in which the user expects them. The sequence must be continuous. The last step must be completion step)
IDREQUEST	The request ID as maintained in MSTCHANNELATS that should be invoked for the step in the wizard.	RRTNC01, RRFCP01, etc.. (Must be proper idrequest maintained in MSTCHANNELATS. Should not have dependency on any other idrequest)
ISMANDATORY	This flag indicates whether the successful completion of the step is mandatory.	Y or N
RES_IDREQUEST	The result request Id. This is the request id which is checked to determine the successful completion of the step.	RRTNC02, RRFCP02, etc.. (Must be proper idrequest maintained in MSTCHANNELATS.)
IDTXN	The transaction ID to which the request id belongs	TNC, FCP, etc.. (Must be a transaction id configured in msttxn)

4.6.2. Names, Descriptions and Skip text

The following configurations need to be done to display the description text to the user for each step:

APPLDATA ENTRY FOR WIZARD_STEP_NAME

To display the name of the step (for example "Step 1: Force Change Password") an entry for dataname WIZARD_STEP_NAME must be made in appldata for appropriate datavalue. The below table explains the values to be entered for each column in APPLDATA:

APPLDATA Table Column Name	Value for WIZARD_STEP_NAME
IDAPP	A1
DATANAME	WIZARD_STEP_NAME
DATAVALUE	<WIZARD ID>.<STEP SEQUENCE NUMBER> (as maintained in WIZARDDETAILS)
IDLANG	Value as per requirement from mstlang.idlang column.
IDDEVICE	Value as per requirement from mstdevice.iddevice column. This column may contain value "**" if the description is same for all devices.
VALUESTRING	The name of the step to be displayed to the user.
LOADFLAG	Y

APPLDATA ENTRY FOR WIZARD_STEP_DESC

To display the description of the step an entry for dataname WIZARD_STEP_DESC must be made in appldata for appropriate datavalue. The below table explains the values to be entered for each column in APPLDATA:

APPLDATA Table Column Name	**Value for WIZARD_STEP_NAME**
IDAPP	A1
DATANAME	WIZARD_STEP_DESC

DATAVALUE	<WIZARD ID>.<STEP SEQUENCE NUMBER> (as maintained in WIZARDDETAILS)
IDLANG	Value as per requirement from mstlang.idlang column.
IDDEVICE	Value as per requirement from mstdevice.iddevice column. This column may contain value "**" if the description is same for all devices.
VALUESTRING	The description of the step to be displayed to the user.
LOADFLAG	Y

APPLDATA ENTRY FOR WIZARD_STEP_SKIP_DESC

To display a warning message to the user in case the user skips a step is maintained against the WIZARD_STEP_SKIP_DESC dataname in appldata table. The below table explains the values to be entered for each column in APPLDATA:

APPLDATA Table Column Name	Value for WIZARD_STEP_NAME
IDAPP	A1
DATANAME	WIZARD_STEP_SKIP_DESC
DATAVALUE	<WIZARD ID>.<STEP SEQUENCE NUMBER> (as maintained in WIZARDDETAILS)
IDLANG	Value as per requirement from mstlang.idlang column.
IDDEVICE	Value as per requirement from mstdevice.iddevice column. This column may contain value "**" if the description is same for all devices.
VALUESTRING	The description of the step to be displayed to the user.
LOADFLAG	Y

4.6.3. Limitations

This section enlists the limitations of the Wizard Framework.

1. Multiple wizards cannot be configured against a combination of ID_ENTITY, USERTYPE, IDCHANNEL and IDREQUEST. Only 1 wizard per request can be configured.
2. The wizard as of now does not provide any means to navigate steps backwards.
3. Each step is committed individually irrespective of other steps. A common confirmation screen cannot be displayed for all steps. However, it does display a summary of all the steps executed.
4. If specific success messages are required to be displayed, the same must be set as result message in the service for the appropriate step.

4.7. Wizard configuration

Oracle FLEXCUBE Direct Banking provides a functionality wherein specific Wizards can be configured. This is an optional configuration to specify the workflow for any given set of transactions.

The following configurations are needed to enable wizards:

4.7.1. Database Configurations

MSTWIZARD

This table holds the base/master configuration for Wizards. The configurations in this table determine the condition for which the Wizard is to be invoked. Following are the columns and the expected values:

Column Name	Description	Expected Values
ID_ENTITY	The Entity ID for which the wizard is to be invoked	F001, B001, B002…. (Must be a correct entity configured in MSTENTITYUSERCHANNELS table)
USERTYPE	The User Type for which the wizard is to be invoked	ECU, EN1… (Must be a proper user type as configured in MSTENTITYUSERCHANNELS table)
IDCHANNEL	The Channel/Device ID for which the wizard is to be invoked	01, 42, 43… (Must be a proper channelis as configured in MSTENTITYUSERCHANNELS table)
IDREQUEST	The Request ID for which the wizard is to be invoked	RRLGN01 (Must be a proper idrequest as maintained in MSTCHANNELATS table)
IDWIZARD	The unique identifier for the wizard. This is referenced in WIZARDDETAILS table.	FLOGIN, FLLOGINRET (Any character string allowed)
START_CONDITION	The fully qualified java class name that defines the condition under which the wizard can be invoked.	This must be the fully qualified name of a Java class that implements the com.iflex.fcat.channels.wizard. WizardVerifier interface. This column may be left *null*. If null no condition would be checked other than matching the ID_ENTITY, USERTYPE, IDCHANNEL and IDREQUEST for the incoming request.

WIZARDDETAILS

Column Name	Description	Expected Values
IDWIZARD	The unique identifier for the wizard.	FLOGIN, FLLOGINRET (Any character string allowed. The same needs to be referenced in MSTWIZARD.IDWIZARD column)
STEPSEQ	The step sequence number of the particular step in the wizard.	0, 1,… (Must be a non-negative integer. The first step must always be 0. All steppseq for a given wizard must be sequential in the order in which the user expects them. The sequence must be continuous. The last step must be completion step)
IDREQUEST	The request ID as maintained in	RRTNC01, RRFCP01, etc.. (Must

	MSTCHANNELATS that should be invoked for the step in the wizard.	be proper idrequest maintained in MSTCHANNELATS. Should not have dependency on any other idrequest)
ISMANDATORY	This flag indicates whether the successful completion of the step is mandatory.	Y or N
RES_IDREQUEST	The result request Id. This is the request id which is checked to determine the successful completion of the step.	RRTNC02, RRFCP02, etc.. (Must be proper idrequest maintained in MSTCHANNELATS.)
IDTXN	The transaction ID to which the request id belongs	TNC, FCP, etc.. (Must be a transaction id configured in msttxn)

4.7.2. Names, Descriptions and Skip text

The following configurations need to be done to display the description text to the user for each step:

APPLDATA ENTRY FOR WIZARD_STEP_NAME

To display the name of the step (for example "Step 1: Force Change Password") an entry for dataname WIZARD_STEP_NAME must be made in appldata for appropriate datavalue. The below table explains the values to be entered for each column in APPLDATA:

APPLDATA Table Column Name	Value for WIZARD_STEP_NAME
IDAPP	A1
DATANAME	WIZARD_STEP_NAME
DATAVALUE	<WIZARD ID>.<STEP SEQUENCE NUMBER> (as maintained in WIZARDDETAILS)
IDLANG	Value as per requirement from mstlang.idlang column.
IDDEVICE	Value as per requirement from mstdevice.iddevice column. This column may contain value "**" if the description is same for all devices.
VALUESTRING	The name of the step to be displayed to the user.
LOADFLAG	Y

APPLDATA ENTRY FOR WIZARD_STEP_DESC

To display the description of the step an entry for dataname WIZARD_STEP_DESC must be made in appldata for appropriate datavalue. The below table explains the values to be entered for each column in APPLDATA:

APPLDATA Table Column Name	Value for WIZARD_STEP_NAME
IDAPP	A1
DATANAME	WIZARD_STEP_DESC
DATAVALUE	<WIZARD ID>.<STEP SEQUENCE NUMBER> (as maintained in WIZARDDETAILS)
IDLANG	Value as per requirement from mstlang.idlang column.
IDDEVICE	Value as per requirement from mstdevice.iddevice column. This column may contain value "**" if the description is same for all devices.
VALUESTRING	The description of the step to be displayed to the user.
LOADFLAG	Y

APPLDATA ENTRY FOR WIZARD_STEP_SKIP_DESC

To display a warning message to the user in case the user skips a step is maintained against the WIZARD_STEP_SKIP_DESC dataname in appldata table. The below table explains the values to be entered for each column in APPLDATA:

APPLDATA Table Column Name	Value for WIZARD_STEP_NAME
IDAPP	A1
DATANAME	WIZARD_STEP_SKIP_DESC
DATAVALUE	<WIZARD ID>.<STEP SEQUENCE NUMBER> (as maintained in WIZARDDETAILS)
IDLANG	Value as per requirement from mstlang.idlang column.
IDDEVICE	Value as per requirement from mstdevice.iddevice column. This column may contain value "**" if the description is same for all devices.
VALUESTRING	The description of the step to be displayed to the user.
LOADFLAG	Y

4.7.3. Limitations

This section enlists the limitations of the Wizard Framework.

5. Multiple wizards cannot be configured against a combination of ID_ENTITY, USERTYPE, IDCHANNEL and IDREQUEST. Only 1 wizard per request can be configured.
6. The wizard as of now does not provide any means to navigate steps backwards.
7. Each step is committed individually irrespective of other steps. A common confirmation screen cannot be displayed for all steps. However, it does display a summary of all the steps executed.
8. If specific success messages are required to be displayed, the same must be set as result message in the service for the appropriate step.

4.8. Extension Audit Logging

Flag `ISAUDITREQUIRED` in table `MSTSERVICES` is used to determine if auditing is required for the respective service. Request and response of business tier is audited in table `ROUTERAUDITLOG`.

ISAUDITREQUIRED = N: Do not audit for the corresponding service.

ISAUDITREQUIRED = Y: Audit for the corresponding service.

4.9. EXTENTION CUSTOM Validations

Oracle FLEXCUBE Direct Banking provides highly customizable validation module that can be quickly customized to use pre-defined validations or incorporate customized validation solution.

Please refer to section on Validation engine for features and details.

The abstract class `com.iflex.fcat.xjava.data.Validator` provides the model for extending validation behavior. This class models the validators to be plugged into the `ValidationEngine`. All implementing validators should conform to this interface.

```
public int validate (
       Hashtable           p_table
,      String              p_value
,      TxnData             p_txndata
,      TxnDataElement      p_txndata_elem
,      Connection          p_con
,      ValidationResult    p_result
)
```

This method should be implemented by the subclasses to provide for the actual validation. The appropriate instance of the validator shall be called by the factory. The method shall return zero (0) in case of a successful validation else it shall return the actual error code. The implementing validator can also add custom error messages to `ValidationResult` reference available.

5. Interfacing with External Systems

Oracle FLEXCUBE Direct Banking provides in-built capability to interface with External Applications such as Oracle Real-Time Decisions, Oracle ATG Web Commerce Live Help On Demand, etc. This section describes the configurations available in FLEXCUBE Direct Banking for such interfacing.

5.1. Integration with Live Help by Oracle ATG

Oracle FLEXCUBE Direct Banking provides options to the business users for interactions with bank officials/call centre executives i.e. Call (Click to Call) facility and Live Help Chat on Demand by integrating with Oracle's Live Help On Demand module of Oracle ATG web Commerce. The existing users have an option to chat with or call the Oracle ATG agents for online assistance.

The following sections describe the configurations required to enable integration with a configured Live Help On Demand account.

💣 The configuration of Oracle ATG Live Help On Demand is not in scope of this document and must be done as per instructions received from Oracle ATG engineers.

5.1.1. Live Help Account Setup

Oracle ATG web commerce's Live Help On Demand provides a account number to the subscribing client. This account number is required for all interactions with Live Help On Demand services. Therefore, the account number needs to be maintained in Oracle FLEXCUBE Direct Banking local database as maintenance.

The following property entry is required in MSTPROPERTIES database table:

Property Name	Property Value
<Entity ID>.LIVE_HELP.ACCOUNTID	The account number as provided by Live Help On Demand.

5.1.2. Live Help Availability Configurations

The Live Help options can be provided to the user as per below configurations:

1. Global : Available for all transactions, except widgets and dashboard.
2. Transaction Level : Available for specific transactions as configured. If transaction level configuration for Live Help is not available, error level configurations are considered, if available.
3. Error Level : Only available for configured errors and warnings in all transactions except widget and dashboard.

The above configurations can be achieved by making appropriate entry against the property name `<Entity ID>.<User Type>.LIVE_HELP.LEVEL` in the `MSTPROPERTIES` database table. Following are the permissible values

LIVE_HELP.LEVEL permissible value	Description
G	Global Level configuration
T	Transaction priority/Error fallback Level configuration
E	Error Level configuration
N	No Live Help needed

Example:

If all user of the Entity Id “B001” belonging to Retail User Segment (User Type EN1) are to be configured such that Live Help is available across all transactions the following configuration would be needed:

```
B001.EN1.LIVE_HELP.LEVEL=G
```

If all user of the Entity Id “B001” belonging to Corporate User Segment (User Type EN1) are to be configured such that Live Help is available across only to specific transactions the following configuration would be needed:

```
B001.EN1.LIVE_HELP.LEVEL=G
```

5.1.3. Transaction Level Configurations

If transaction level configurations for Live Help are required, the transactions are needed to be categorized into “Live Help Modules”. Each “Live Help Module” can have multiple transactions in it. Each “Live Help Module” can have a separate “Department” of Agents who can handle queries related the “Live Help Module”.

To configure a “Live Help Module” for a specific transaction, the column LIVEHELPMODULEID in MSTTXN should be updated with an ID string. This string can be any alpha-numeric string.

Further, after configuring the “Live Help Module” for transactions, each module would need to be configured for integration with the Live Help On Demand service. The database table LIVEHELPCONFIG holds the detailed configuration of such mapping. The details of the database table are as below:

Column Name	Description	Expected Value
ID_ENTITY	The Entity ID for which the Live Help Module configuration is to be maintained.	F001, B001, B002…. (Must be a correct entity configured in MSTENTITYUSERCHANNELS table)
USERTYPE	The User Type for which the Live Help Module configuration is to be maintained.	ECU, EN1… (Must be a proper user type as configured in MSTENTITYUSERCHANNELS table)
IDCHANNEL	The id channel for which the Live Help Module configuration is to be maintained.	01, 11… (Must be a proper id channel as configured in MSTENTITYUSERCHANNELS table)
IDMODULE	Indicates the Live Help Module for which the configuration is to be maintained	No validations on this column. However, the module id should be the same as module id configured in LIVEHELPMODULEID column of MSTTXN database table.
IDCALLTEMPLATE	The template ID as generated in Live Help On Demand for the “Click to Call” action for the department to which the module id is to be mapped.	This value can either be null or a valid numeric template id. If null it is assumed that “Click to Call” functionality is not required.
IDCHATTEMPLATE	The template ID as generated in Live Help On Demand for the “Click to Chat” action for the department to which the module id is to be mapped.	This value can either be null or a valid numeric template id. If null it is assumed that “Click to Chat” functionality is not required.
ISPAGEPEEKENABLED	Indicates whether the page peek functionality is enabled for the current configuration.	Either ‘Y’ or ‘N’.

ISCONFIGENABLED	Indicates whether the current configuration is enabled.	Either 'Y' or 'N'.
IDLANG	Indicates the language in which the user has logged in and therefore correspondingly the language of the department to which the call/chat is to be forwarded.	The 3 character language id as maintained in MSTLANG database table.

For Error Level Fallback Configuration kindly refer the section 11.10.4

5.1.4. Error Level Configurations

If error level configurations for Live Help are required, the transactions are needed to be categorized into "Live Help Modules". Each "Live Help Module" can have multiple error codes configured in it. Each "Live Help Module" can have a separate "Department" of Agents who can handle queries related the "Live Help Module". The Live Help Module configured for errors can be the same as the one configured for transactions if the same department of agents would be handling such service requests.

The database table LIVEHELPERRORCONFIG holds the mapping of error codes against the module id. The details of the database table are as below:

Column Name	Description	Expected Value
ID_ENTITY	The Entity ID for which the Live Help Module configuration is to be maintained.	F001, B001, B002…. (Must be a correct entity configured in MSTENTITYUSERCHANNELS table)
USERTYPE	The User Type for which the Live Help Module configuration is to be maintained.	ECU, EN1… (Must be a proper user type as configured in MSTENTITYUSERCHANNELS table)
IDCHANNEL	The id channel for which the Live Help Module configuration is to be maintained.	01, 11… (Must be a proper id channel as configured in MSTENTITYUSERCHANNELS table)
IDMODULE	Indicates the Live Help Module mapped for the error code.	No validations on this column.
IDMESSAGE	The error code or warning code for which the configuration is to be maintained.	This value should be a valid error code/warning code as maintained in IDMESSAGE field in APPLICATIONMESSAGE database table.
IDLANG	Indicates the language in which the user has logged in and therefore correspondingly the language of the department to which the call/chat is to be	The 3 character language id as maintained in MSTLANG database table.

	forwarded.	