Oracle FLEXCUBE Direct Banking

System Handbook – Volume V – Host Integration Layer
Release 12.0.3.0.0

Part No. E52543-01

April 2014

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. Intended Audience

This System Handbook (Volume IV – Business Services 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://support.us.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 12.0.3.0.0, 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 III
- Oracle FLEXCUBE Direct Banking System Handbook – Volume IV
- Oracle FLEXCUBE Direct Banking System Handbook – Volume V

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 (💣).

3. Integration Tier

The Integration Tier comprises of the FLEXCUBE application integration platform. The Integration tier will interface with the backend systems or middleware synchronously, asynchronously or via batch mode of integration.

The integration tier can support for multiple communication protocols like MQ/JMS, Web Services, RMI/IIOP, TCP/IP and message formats like XML, ISO 8583, ASCII etc.

In case of point to point communication with the backend systems the integration tier can perform message transformation, message routing and orchestration. In case the integration with backend systems is via EAI platform / middleware then message transformation, message routing, orchestration can be performed by EAI platform / middleware.

3.1. Host Interface Framework

This chapter discusses the details of the Host Interface Framework available for interfacing with any host system. The framework consists of the below components:

1. HostAdapterManager : This is a static java class (`com.iflex.fcat.services.hostinterface.HostAdapterManager`) responsible for invoking individual host adapters. This is a final class cannot be extended.
2. HostAdapterHelper: This static class (`com.iflex.fcat.services.hostinterface.HostAdapterHelper`) provides helper methods for use in HostAdapterManager and adapter implementations. This is a final class cannot be extended.
3. MSTHOSTINTERFACE: This database table holds the meta information of all host interface invocations available in the system. Entries can be added to this table to implement more host interfaces as required for the implementation. The details of this table are discusses in section 3.1.1 below.
4. HostAdapter : This is the interface that must be implemented by all adapters for communication with any host system. The section 3.1.2 further discusses some default out-of-box interface adapters.

3.1.1. Host Interface Configurations

All configurations available for host integration are available in the database table MSTHOSTINTERFACE. Below are the details of the available configurations:

Column Name	Description	Datatype
ID_ENTITY	The entity for which the configuration is to be maintained	VARCHAR2 (5)
IDHOST	The unique host identifier for the host which is to be invoked. Below values are	VARCHAR2 (5)

	available out-of-box: FCDB : Third Party Host LOCAL : Refers local FCDB tables as an external host. UB11X : FCUBS	
IDREQUEST	The request ID for which the interface is to be invoked. This may be any alpha numeric value to uniquely identify the request.	VARCHAR2(50)
VERSION	The version of the interface.	NUMBER(255)
REQUEST_IFACE_MAPPER	The fully qualified class name of the mapper which converts the request DTO available from the service tier into a request DTO understandable by the adapter. This is a non-mandatory field. If no mapper is specified, the request available from the service is assumed to be the same as required by the adapter.	VARCHAR2 (255)
RESPONSE_IFACE_MAPPER	The fully qualified class name of the mapper which converts the adapter's response DTO into a request DTO understandable by the service. This is a non-mandatory field. If no mapper is specified, the adapter's response would be passed as-is to the service.	VARCHAR2 (255)
INTERFACE_IMPL	The fully qualified name of the adapter implementation. This must implement the "HostAdapter" interface. Please refer section 3.1.2 for details on out-of-box available adapters.	VARCHAR2 (4000)
ISENABLED	Specifies if the interface is enabled.	CHAR (1)
ADTNL_PARAMS	Holds any additional configurations that might be required for the invocation of the adapter.	VARCHAR2 (4000)
IDINTERFACE	An identifier to uniquely identify the interface	VARCHAR2 (20)
REF_ID_ENTITY	Comma separated list of entities to which the same interface is to be copied.	VARCHAR2 (300)

3.1.2. Out-of-box adapter implementations

For ease of use and standardization, FCDB provides a few out-of-box implementations of adapters. The most relevant adapter from customization standpoint is FlexmlHostAdapter and ThirdPartyHostAdapter. FlexmlHostAdapter (`com.iflex.fcat.services.hostinterface.impl.FlexmlHostAdapter`) provides interfacing over multiple communication channels namely Message Queues, EJB, and Web Services. Below are the additional parameters available for each of the communication channels:

1. Message Queue

Parameter Name	Description	Available Values
FLEXML.MODE	Specifies the communication mode for the adapter.	Q – Message Queue
FLEXML.MQ.PREFIX	The prefix to be used for looking up properties from application properties (mstproperties or fcat.properties)	
RESPONSE.CLASS.NAME	The fully qualified class name of the response DTO the adapter is expected to return.	
FLEXML.REQUSET.XSL	The XSL to be used to transform the marshaled XML format of the DTO into format readable by the host. If not specified, the marshaled XML format of the DTO is assumed to be the input required by the host.	
FLEXML.RESPONSE.XSL	The XSL to be used to transform the XML message available from the host into marshaled XML format of the DTO. If not specified, the marshaled XML format of the DTO is assumed to be available from the host.	
REQ.XSD.VALIDATION.FLAG	Specified whether the request XML is to be validated against an XSD.	Y – Yes N – No
RES.XSD.VALIDATION.FLAG	Specified whether the response XML is to be validated against an XSD.	Y – Yes N – No
IS.HOST	Specifies whether commit is required within the adapter	Y – Yes N – No
FLEXML.RETRY.FLAG	Specifies whether the interface is to be retries in case of technical failures.	Y – Yes N – No
FLEXML.EODCHECK	Specifies EOD check before sending transaction across to host	T – Terminate if past EOD Y – Tank the request until EOD is under process N – No EOD check required.
FCDB.REQ.XSL.NAME	The XSL to be applied on the delayed response to invoke a service in FCDB for updation of data.	
HOST.AUDIT.ENABLED	Enable auditing of the request	Y – audit N – do not audit

2. EJB

Parameter Name	Description	Available Values

FLEXML.MODE	Specifies the communication mode for the adapter.	E – EJB
FLEXML.EJB.REMOTE.CLASS	The remote class for the EJB	
FLEXML.EJB.HOME.CLASS	The home class of the EJB	
FLEXML.EJB.REMOTE.METHOD	Method to be invoked on the EJB	
FLEXML.EJB.JNDI.NAME	JNDI name of the EJB	
RESPONSE.CLASS.NAME	The fully qualified class name of the response DTO the adapter is expected to return.	
FLEXML.REQUSET.XSL	The XSL to be used to transform the marshaled XML format of the DTO into format readable by the host. If not specified, the marshaled XML format of the DTO is assumed to be the input required by the host.	
FLEXML.RESPONSE.XSL	The XSL to be used to transform the XML message available from the host into marshaled XML format of the DTO. If not specified, the marshaled XML format of the DTO is assumed to be available from the host.	
REQ.XSD.VALIDATION.FLAG	Specified whether the request XML is to be validated against an XSD.	Y – Yes N – No
RES.XSD.VALIDATION.FLAG	Specified whether the response XML is to be validated against an XSD.	Y – Yes N – No
IS.HOST	Specifies whether commit is required within the adapter	Y – Yes N – No
FLEXML.RETRY.FLAG	Specifies whether the interface is to be retries in case of technical failures.	Y – Yes N – No
FLEXML.EODCHECK	Specifies EOD check before sending transaction across to host	T – Terminate if past EOD Y – Tank the request until EOD is under process N – No EOD check required.
HOST.AUDIT.ENABLED	Enable auditing of the request	Y – audit N – do not audit

3. Web Services

Parameter Name	Description	Available Values
FLEXML.MODE	Specifies the communication mode for the adapter.	W – Web Services
FLEXML.WS.SERVICE.CLASS	The web service's stub class	

	generated using the method mentioned in following section	
FLEXML.WS.PORT.CLASS	The port class generated for the web service	
FLEXML.WS.OPERATION.NAME	Operation name of the web service	
FLEXML.WS.OPERATION.PARAMS	Comma separated list of data types of the parameters to the service.	
FLEXML.INTERCEPTOR	Fully qualified class name of the intercepter implementing com.iflex.fcat.hostinterface.impl.HostIntercepter. This class is responsible for converting the request DTO to the parameter objects. It also converts the output object available from the host into a response DTO.	
RESPONSE.CLASS.NAME	The fully qualified class name of the response DTO the adapter is expected to return.	
FLEXML.REQUSET.XSL	The XSL to be used to transform the marshaled XML format of the DTO into format readable by the host. If not specified, the marshaled XML format of the DTO is assumed to be the input required by the host.	
FLEXML.RESPONSE.XSL	The XSL to be used to transform the XML message available from the host into marshaled XML format of the DTO. If not specified, the marshaled XML format of the DTO is assumed to be available from the host.	
REQ.XSD.VALIDATION.FLAG	Specified whether the request XML is to be validated against an XSD.	Y – Yes N – No
RES.XSD.VALIDATION.FLAG	Specified whether the response XML is to be validated against an XSD.	Y – Yes N – No
IS.HOST	Specifies whether commit is required within the adapter	Y – Yes N – No
FLEXML.RETRY.FLAG	Specifies whether the interface is to be retries in case of technical failures.	Y – Yes N – No
FLEXML.EODCHECK	Specifies EOD check before sending transaction across to host	T – Terminate if past EOD Y – Tank the request until EOD is under

		process N – No EOD check required.
HOST.AUDIT.ENABLED	Enable auditing of the request	Y – audit N – do not audit

4. 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.

4.1. 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.

4.1.1. 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.

4.1.2. 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
```

4.1.3. 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	This value can either be null or a valid numeric template id. If null it is assumed that “Click to Chat” functionality is

	module id is to be mapped.	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

4.1.4. 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	ECU, EN1… (Must be a proper user type as configured in MSTENTITYUSERCHANNELS

	maintained.	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 forwarded.	The 3 character language id as maintained in MSTLANG database table.

4.2. Targetted and Location Based Offers

Oracle FLEXCUBE Direct Banking can display personalized offers and advertisements to the users in the online banking channels – Internet Banking, Application Based Mobile Banking, Browser Based Mobile Banking and Tablets. Oracle FLEXCUBE Direct Banking can integrate with third party systems like Oracle Real Time Decision application for determining the applicable offers. Oracle FLEXCUBE Direct Banking also provides an internal offer maintenance engine in case there are no offer generation applications available.

- The internal offer maintenance engine is merely an offer maintenance system and not an offer generation system. It does not generate offers based on the user's details and activities. The offers maintained in the system are to be uploaded via Static File Upload Utility. Please refer to the document Oracle_FLEXCUBE_Direct_Banking_Static_Data_Upload_Utility for further details on how to upload offers to the offer maintenance system.

- The internal offer maintenance engine is only applicable for Targeted Offers and is not available for Location Based Offers.

To configure the offer engine for Targeted Offers, the following configurations are needed:

Channel	Property
Internet	<ENTITYID>.<USERTYPE>.ADS.HOST.NAME
Mobile Application	<ENTITYID>.<USERTYPE>.ADM.HOST.NAME
Mobile Browser	<ENTITYID>.<USERTYPE>.ADM.HOST.NAME

The available values for the above configurations are as below:

Property Value	Description
RTD	Oracle Real Time Decisions as the offer generation engine.
LOCAL	Oracle FLEXCUBE Direct Banking as the offer maintenance engine.
FCDB	Any Third Party Offer Generation/Maintenance engine capable of communicating over EJB/JMS channels.

To configure the offer engine for Location Based Offers, the following configurations are needed:

Channel	Property
Internet	- Functionality Not Available -
Mobile Application	<ENTITYID>.<USERTYPE>.ADL.HOST.NAME
Mobile Browser	- Functionality Not Available -

The available values for the above configurations are as below:

Property Value	Description
RTD	Oracle Real Time Decisions as the offer generation engine.
FCDB	Any Third Party Offer Generation/Maintenance engine capable of communicating over EJB/JMS channels.

To configure the Content Management System for all offer engines for Targeted and Location Based offers, the following configurations are needed:

Channel	Property
Internet	<ENTITYID>.<USERTYPE>.FETCHADDET.HOST.NAME
Mobile Application	<ENTITYID>.<USERTYPE>.FETCHADDET.HOST.NAME
Mobile Browser	<ENTITYID>.<USERTYPE>.FETCHADDET.HOST.NAME

The available values for the above configurations are as below:

Property Value	Description
LOCAL	Oracle FLEXCUBE Direct Banking as the offer maintenance engine.

To utilize a third party Content Management System, a new Host Name would need to be defined along with the development of appropriate adapters.

ⓘ For installation and configuration of Oracle Real Time Decisions, kindly refer the document "**Oracle_FLEXCUBE_Direct_Banking_Oracle_RTD_Installer_UserGuide**".

4.3. Private Wealth Management

Oracle FLEXCUBE Direct Banking – Oracle FLEXCUBE Private Banking integration allows Oracle FLEXCUBE Direct banking business user to view and manage their investment portfolio available with Oracle FLEXCUBE Private Banking application using Single Sign On (SSO) with Oracle FLEXCUBE Direct Banking application. Hence, the business users shall be able to access FCPB application through FCDB login. This functionality is only available to the Retail Customers of the bank.

ⓘ For configuring Oracle FLEXCUBE Direct Banking for interaction with Oracle FLEXCUBE Private Banking, kindly refer the document "**Oracle_FLEXCUBE_Wealth_Management_Interface**".

4.4. Integration with Oracle IPM for Mailbox attachment

Oracle FLEXCUBE Direct Banking – Oracle IPM Integration allows attaching document in mails to external DMS system. The same document will be accessible to host from Oracle IPM Tools acting as external DMS system. DMS system will be acting as repository for Mailbox attaches documents. This is in case, when the mailbox acting as Interactive module. And In case of non-interaction Mailbox, attachments are saved into the database.

4.5. Integration of virus scanning tool for files to be attached

Oracle FLEXCUBE Direct Banking – Allows to integrate any virus scanning tool to scan presence of malicious content in file to be uploaded through any transaction.

This is configurable through properties.

There are 2 properties for this functionality is as follows

1) FILEUPLOAD.SCAN.REQUIRED

2) FILEUPLOAD.SCAN.VALIDATOR

FILEUPLOAD.SCAN.REQUIRED:- This flag provides the information to the system whether to enable/disable virus scanning functionality to the uploading files.

FILEUPLOAD.SCAN.VALIDATOR:- This property provides the plug-in class name which would route your file content to the tool for scanning. This class must be an implementation of Interface `com.iflex.fcat.gui.FileScanValidator`. The implementer is required to create a new Java class implementing the above mentioned interface. The newly created Java class needs to be plugged in using this property for Virus Scan validation..

4.6. FLEXCUBE UBS Core Banking System

Oracle FLEXCUBE Direct Banking – Oracle FLEXCUBE Universal Banking integration allows Oracle FLEXCUBE Direct banking business user

4.7. Integration with Third party system

Oracle FLEXCUBE Direct Banking – Oracle FLEXCUBE Universal Banking integration allows Oracle FLEXCUBE Direct banking business user