Oracle FLEXCUBE Universal Banking® 12.0
Development Overview Guide

Release 1.0

May 2012

Oracle Part Number E51465-01

Contents

1 Preface

This Development Overview document provides the bird's eye view of FLEXCUBE UBS Application development. It touches the concepts, frameworks, tools required and documents available for guidance.

1.1 Audience

This Developer Overview book is intended for authorized FLEXCUBE UBS Application Developers who are expected to perform the following task:

- To develop a Function ID(User Interface Screen)
- To develop a Web Service
- To develop a Notification
- To Extend FLEXCUBE functionality using with extensibility
- To Interface FLEXCUBE with external systems using Generic Interface
- To upload data into FLEXCUBE using upload adaptors
- To use FLEXCUBE UBS framework tools
- To develop BIP Reports
- To develop OBIEE Repository files
- To debug FLEXCUBE at run time
- To Analyze and fix FLEXCUBE UBS bugs

To Use this manual, you need conceptual and working knowledge of the below:

Table 1.1 – Proficiency and resources

Proficiency	*Resources*
FLEXCUBE Functional Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Technical Architecture	Training programs from Oracle Financial Software Services.
Working knowledge of Web based applications	
Working knowledge of Oracle Database	Oracle Documentations
Working knowledge of PLSQL developer	Respective vendor documents
Working knowledge of PLSQL & SQL Language	
Working knowledge of XML files	

1.2 Related Documents

Refer the below documents for more information on FLEXCUBE UBS Application development.

1. FCUBS-FD01-01-01-Development Overview Guide
2. RAD
    a. FCUBS-FD02-01-01-RAD Getting Started
    b. FCUBS-FD02-02-01-RAD Function ID Development Volume 1
    c. FCUBS-FD02-02-01-RAD Function ID Development Volume 2
    d. FCUBS-FD02-03-01-RAD Web Service Development
    e. FCUBS-FD02-04-01-RAD BIP Report Integration
    f. FCUBS-FD02-05-01-RAD Notification Development
3. Extensibility
    a. FCUBS-FD03-01-01-Extensibility Getting started
    b. FCUBS-FD03-02-01-Extensibility Reference Guide
    c. FCUBS-FD03-03-01-Extensibility By Example Volume 1
    d. FCUBS-FD03-03-02-Extensibility By Example Volume 2
4. Interface
    a. FCUBS-FD04-01-01-Interface Getting started
    b. FCUBS-FD04-02-01-Generic Interface Configuration Guide
    c. FCUBS-FD04-03-01-Upload Adapter Development Guide
5. Tools
    a. FCUBS-FD05-01-01-Tools-Getting Started
    b. FCUBS-FD05-02-01-RAD-Reference
    c. FCUBS-FD05-02-02-RAD-Installation and Setup
    d. FCUBS-FD05-03-01-DDL-Reference
    e. FCUBS-FD05-04-01-TrAX-Reference
6. Support
    a. FCUBS-FD06-01-01-Support Getting started
    b. FCUBS-FD06-02-01-Support By Example
7. Reports
    a. FCUBS-FD07-01-01-Report Getting started
    b. FCUBS-FD07-02-01-BIP Report Development Guide
    c. FCUBS-FD07-03-01-OBIEE repository Development Guide
8. Data model
    a. FCUBS-FD08-01-01-Data Model getting started

1.3 Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements (for example, menus and menu items, buttons, tabs, dialog controls), including options that you select.

italic	*i*talic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates language and syntax elements, directory and file names, URLs, text that appears on the screen, or text that you enter.

2 Introduction

2.1 How to use this Guide

The information in this document includes:

- Chapter 2, "Introduction"
- Chapter 3, "FLEXCUBE UBS Development – Introduction"
- Chapter 4, "FLEXCUBE UBS Application Developer Documents"
- Chapter 5, "Developer Glossary"

3 FLEXCUBE UBS Development - Introduction

FLEXCUBE UBS Application development consists of three parts:

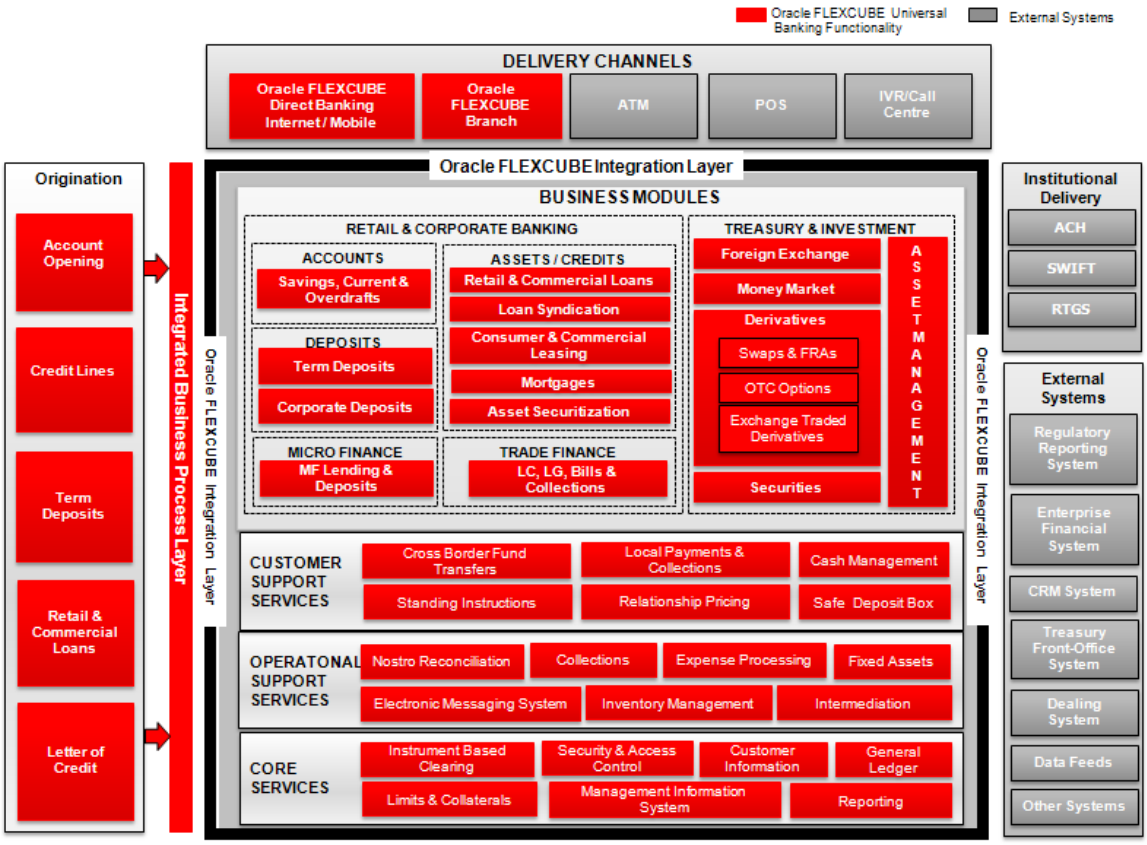
- Framework development
- Tools development
- Application components development using Framework/Tools

This document and associated documents are intended to guide FLEXCUBE UBS "Application component development".

3.1 FLEXCUBE UBS Functional architecture overview

The given below diagram provides the functional architecture of the FLEXCUBE UBS. Refer the respective FLEXCUBE UBS user manuals to know functionality.

Fig 3.1 -FLEXCUBE UBS Functional architecture

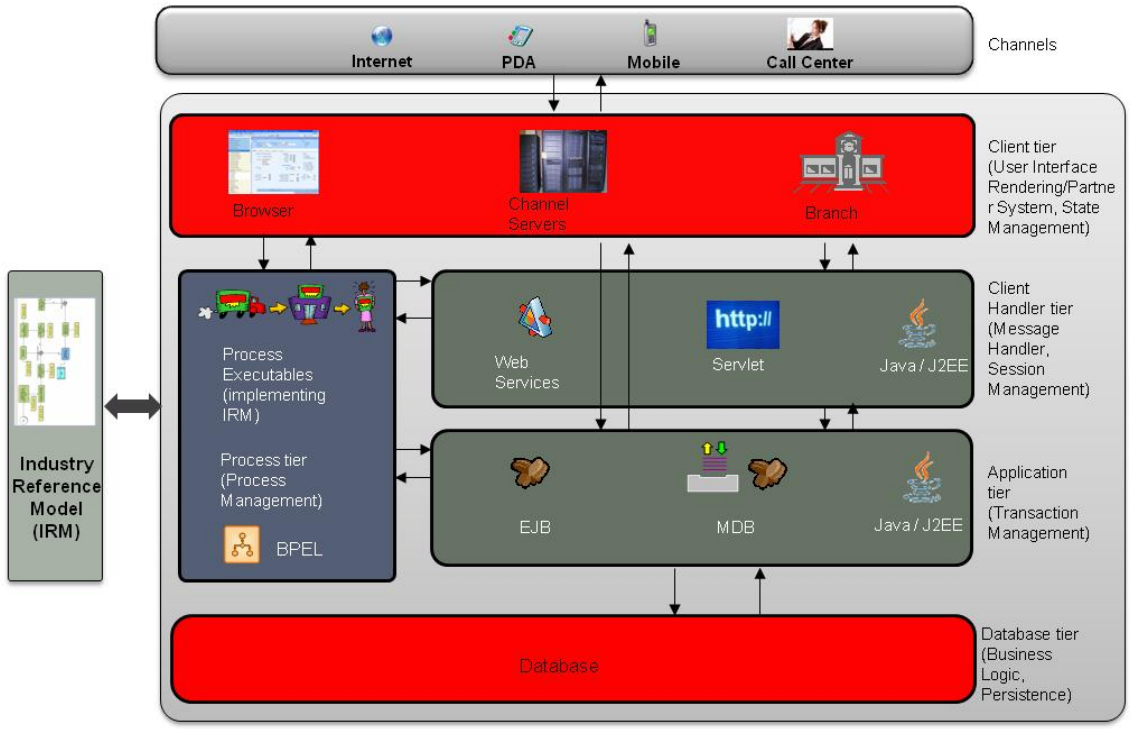


3.2 FLEXCUBE UBS Technical architecture overview

The diagram depicted below represents the FLEXCUBE technical architecture and it consists of the following tiers:

- User Interface tier
- Process tier
- Application and Integration tier
- Database tier

Fig 3.2 -FLEXCUBE UBS Technical architecture



3.2.1 **User Interface tier**

The user interface of the application is light-weight in nature and based on JavaScript and XML. The communication between the browser and the web server is using XML. The rendering is done on the client using XSLT. The user interface is configurable. The screen can be easily adapted to different languages.

3.2.2 **Process tier**

Oracle FLEXCUBE provides for processes to be developed around the natively provided application. One can define processes using Oracle BPEL Process Manager and integrate the same into the application's user interface framework. When deployed in a process centric model, Oracle FLEXCUBE provides a task-based user interface. By default, Task based UI is offered for the branch platform.

3.2.3 **Application and Integration tier**

Oracle FLEXCUBE does not differentiate partner channels from its own native user interface when it comes to data processing. The Application and Integration tier provides the message handling, session management (for the native user interface) and transaction management in the application.

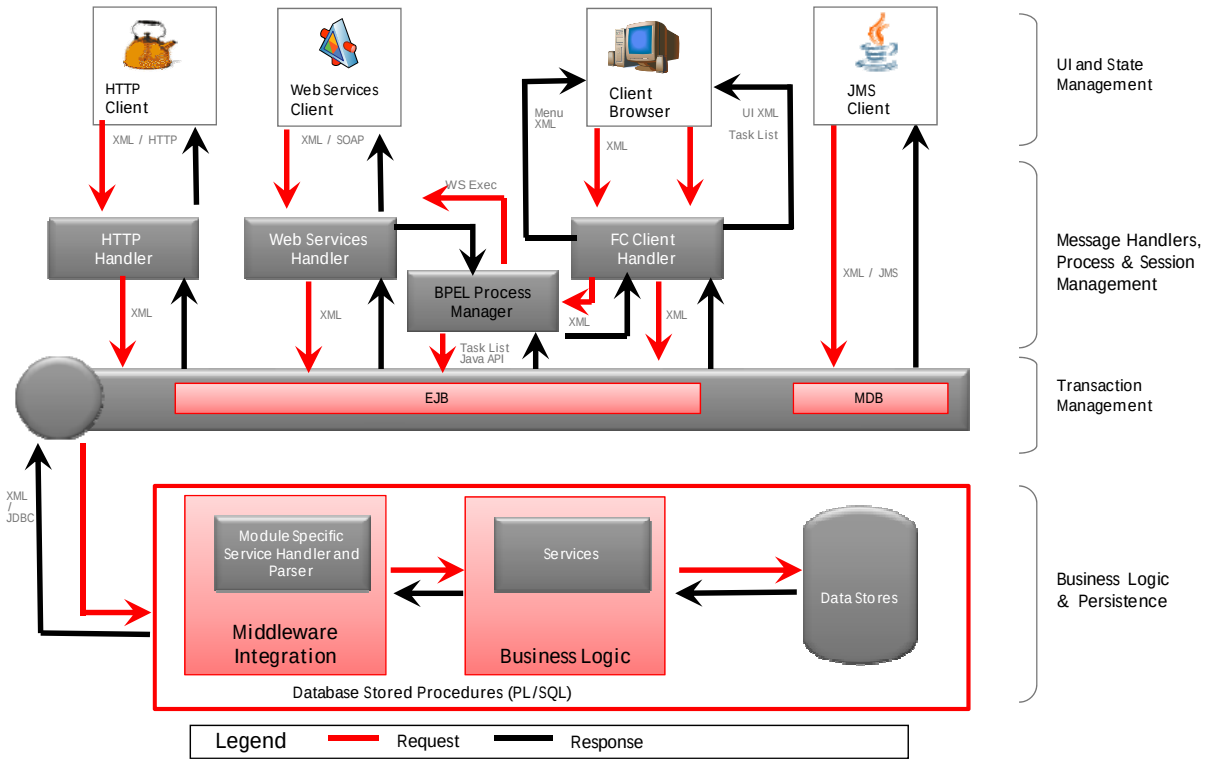
3.2.4 **Database tier**

The back-end is a relational database management system i.e. Oracle 11g. The database tier ensures integrity of data and also provides business logic written mainly in Oracle PL/SQL from the Oracle database.

3.3 FLEXCUBE UBS data flow

The below self explanatory diagram represents the FLEXCUBE UBS data flow at run time.

Fig 33 - FLEXCUBE UBS Data flow



3.4 FLEXCUBE UBS Framework

FLEXCUBE UBS Application consists following frameworks to develop the various parts of Application.

3.4.1 User Interface framework

This is used to develop and render various FLEXCUBE UBS interface (Screens aka Function ID). This consists following sub parts:

- Maintenance
- Online
- Report
- Batch
- ELCM
- Notification
- Branch Screen
- Process flow

3.4.2 **Gateways**

This framework used to develop various gateway components and support wide integration mechanism. This consists of following sub parts:

- Generic XML Gateway
  - EJB
  - MDB
  - Servlet
  - Web Service
  - Notifications
- Generic ASCII Interface
- EMS Gateway
- Switch Gateway

3.4.3 **Extensible**

This framework allows developing FLEXCUBE UBS extensions

3.4.4 **Branch workflow**

This framework used to develop workflow based branch screens.

3.4.5 **Reports**

This framework allows reports development in FLEXCUBE UBS Apps. It consists of the following sub parts:

- BIP
- OBIEE rpd

3.4.6 **BPEL process flows**

This framework used to develop the process flows that are centric to SOA architecture.

3.5 FLEXCUBE UBS Application components & Tools to be used

This section describes the FLEXCUBE UBS components and tools used to develop the components

Table 3.1 – Framework, components and Tools

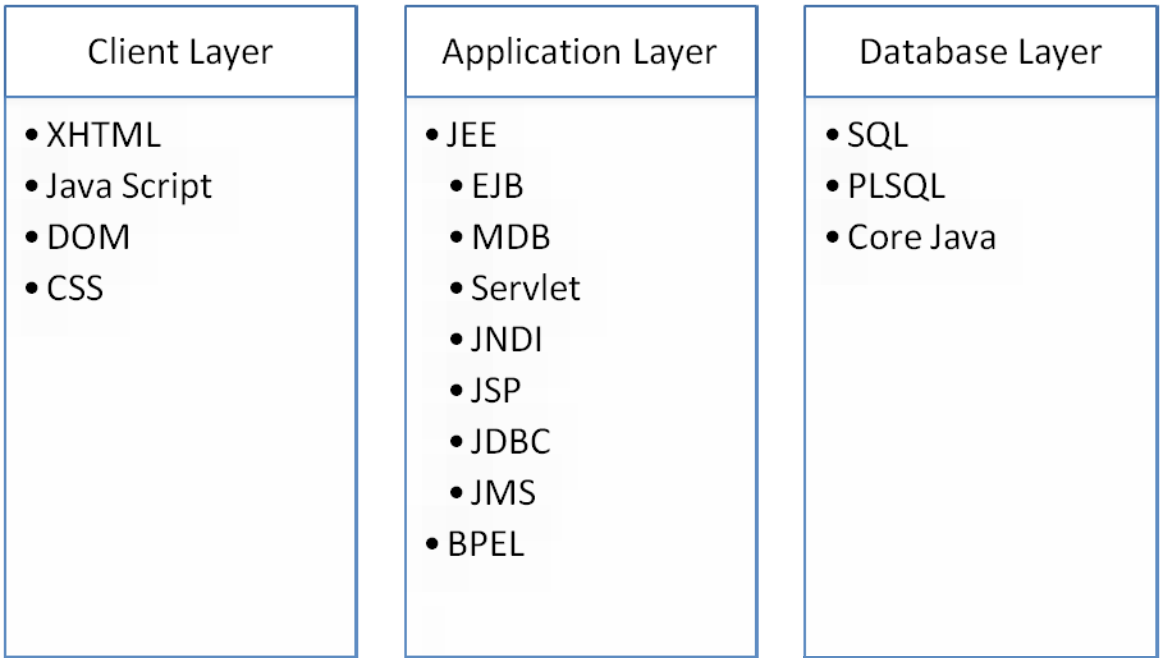
Framework	*Component*	*Tools*
User Interface	Maintenance	RAD
	Online	RAD
	Report	RAD
	Batch	RAD
	ELCM	RAD
	Notification	RAD
	Branch Screen	RAD
	Process flows screen	RAD
Gateway	Web service	RAD, TrAX

	Gateway XML message	RAD, TrAX
	Notification	RAD,TrAX
	ASCII Generic Interface	FLEXCUBE UBS
	Upload Adapter	FLEXCUBE UBS
Reports	BIP Canned reports	RAD, BI Publisher
	OBIEE Meta data repository	RAD, OBIEE Suite
Process flows	Process flows	RAD,JDeveloper,Oracle BPEL process manager

3.6 FLEXCUBE Programming Language Overview

FLEXCUBE UBS Application uses the following programming languages for each layer in technical architecture

Fig 3.4 -FLEXCUBE UBS programming languages



3.7 FLEXCUBE Data Model

FLEXCUBE Data model is available for select modules. This helps to get the following

- ER relationships of FLEXCUBE UBS Tables
- Table comments
- Column comments with enumeration list

FLEXCUBE UBS Data model helps in understanding the Database design and assist to create specific report development and extensibility changes.

3.8 FLEXCUBE UBS Object Naming Conventions

It is essential to know the naming conventions to create FLEXCUBE UBS application objects. This section helps to understand existing objects naming conventions and to create new ones.

3.8.1 **Module**

Every database object names start with the two-character module codes. The below list provides possible module codes.

Table 3.2 – FLEXCUBE UBS Module code list

Module Code	**Module Name**
AC	Accounting
AE	Auto End of Day
AM	Asset Management
BC	Bills and Collections
BL	Billing
BR	Brokerage
CA	CASA
CD	Corporate Deposits
CF	The ICCF
CG	Clearing
CI	Islamic Financing
CL	Retail Lending
CN	Collections
CO	Core Services
CS	Core
CV	Conversion
CY	Currency
DA	Discount Accrual
DD	Demand Draft
DE	Data Entry
DL	Deposit Locker
DV	Derivatives
DX	Data Extraction
ED	Exchange Traded Derivatives
EI	End of Cycle
EN	EXCHANGE DERIVATIVES
EP	ETD Portfolio Product
EP	ETD Portfolio Product
FA	Fixed Assets
FI	Islamic Assets
FR	FLEXCUBE Reporting System
FS	FX Settlements
FT	Funds Transfer
FX	Foreign Exchange

Module Code	Module Name
GA	Cost Allocation
GE	Global Exposure (ELCM)
GI	Generic Interface
GL	General Ledger
GW	Gateway
IA	Islamic Asset Management
IB	Islamic Bills and Collections
IC	Interest And Charges
ID	Islamic Derivatives
IF	Interfaces
IL	Integrated Liquidity Management
IN	Intermediary
IS	Settlement Instructions
IV	Inventory
LC	Letters Of Credit
LD	Loans and Deposits
LE	Leasing
LI	Islamic Letters Of Credit
LL	Participant Tranches and Drawdown
LM	Limits
LN	Loan Syndication
LQ	Receivable Liquidation
LS	Syndication Loans and Commitments
MB	Millionaire Certificates
MC	Islamic Money Market
MG	Margin Management
MI	Management Information System
MM	Money Market
MO	Mortgages
MS	Messaging
NR	Reconciliation System
NT	Netting Across Modules
OB	Security Management System
OP	Branch Vault
OR	Origination
OT	OTC Options
PC	Local Payments
PD	Post Dated Cheques
RE	Reconciliation System
RF	Retail Funds Transfer
RP	Reporting System
RT	Retail Teller
SD	Securities Deal Module
SE	SECURITIES
SF	Structure Deposit
SI	Standing Instruction
SL	Salary Credit

Module Code	Module Name
SM	Security Management System
SP	Securities Portfolio Module
SR	Securities Repo and Reverse Repo
SS	Securities Security Module
ST	Static Maintenance
SV	Signature Verification
SW	Switch Interface
SZ	Securitization of Loans
TA	The Tax
UD	UDF
UP	Utility Payments
VP	Vendor Payment
WB	FLEXCUBE Web Branch
XP	Expense Processing

3.8.2 **Function IDs**

Function IDs created in RAD need to follow the below naming convention:

<*two character module*><*Type*><*functionality*>

<Type> is as follows:

Table 3.3 – Function ID Type list

Third character	*Type*
D	Detail
S	Summary
R	Report
C	Call form
N	Notification
A	Authorization

Example:

- *STDCIF* - *Detailed screen Customer information*
- *STSCIF* – *Summary screen Customer information*
- *ACRJRNAL* – *Report Screen Journal*
- *CONCUSAD* - *Notification of customer address on core module*
- *1001* – *Web branch screen CashWithdrawal*

3.8.3 **Table Names**

- All tables are divided into 3 categories
  - Maintenance Tables
    - Tables that have a front-end form through which data is collected
  - Internal Tables

        - Tables that have no front-end UI for data collection. Their data comes through SQL statements that are executed by program units (either from the back-end or the front-end).

    - Temporary Work Tables
        - Tables that are much like Internal tables. They store data that are required for a short, definite period of time, typically, lasting only as long as the routine that created it is running the given task. These are, generally, cleared automatically after they serve the purpose they were created for. Contrary to the common inference of the table name, these tables are not temporary; rather, the data they contain are.

- The 3rd and 4th characters of the table name identify the type of table
    - "TM" for Maintenance Tables
    - "TB" for Internal Tables
    - "TW" for Temporary (Work) Tables
- All table names have '_' (underscore) as the fifth character.
- Characters from 6th position onwards are used to identify the content of the table
- E.g. Maintenance Table - STTM_CUSTOMER, STTM_CUST_ACCOUNT
    - Internal Table - STTB_ACCOUNT, CSTB_CONTRACT
    - Temp. Work Table - ICTW_IS_VALS

3.8.4 **Package Names**

- Package names generally begin with the module code they belong to.
- All server package names have "PKS" as the 3rd , 4th and 5th characters
- Client-side packages will contain "PKCS", alternatively
- Examples:
    - BCPKS_CONTRACT - Package containing Bills contract related units
    - CFPKS_SERVICES - Package containing ICCF units
    - ACPKS - Package containing Accounting-related units
    - SMPKCS - Front-end package containing SMS-related units

3.8.5 **Views**

- All view names have "VW" as the 3rd and 4th character
- All view names have '_' (underscore) as the 5th character.
- Characters from 6th position onwards are used to identify the content of the view
- Example
    - LCVW_UPLOAD_MASTER - View used to access LC upload data
    - ACVW_All_AC_Entries - View to access all accounting entries

3.8.6 **Triggers**

- All trigger names have "TR" as the 3rd and 4th character
- All trigger names have '_' (underscore) as the 5th character.
- Characters from 6th position onwards are used to identify the purpose of the trigger
- Example
    - CYTR_RATES_HISTORY_UPD - Trigger to update currency rates history

3.8.7 **Synonyms**

All tables, packages and views are referred only through synonyms in code. Synonym names are formed by adding the character "S" after the module and object type identifier

Example

- STTMS_CUSTOMER - Synonym for table STTM_CUSTOMER
- STTBS_ACCOUNT - Synonym for table STTB_ACCOUNT
- BCPKSS_CONTRACT - Synonym for package BCPKS_CONTRACT
- STVWS_CUST_ACCOUNT - Synonym for view STVWS_CUST_ACCOUNT

3.8.8 **File extensions**

FLEXCUBE UBS Application consists of following file extension types.

Table 3.4 – File extensions list

File extension	*Purpose*	*Tools*
ddl	Table DDL file	DDL Tool
inc	INC file	DDL Tool
seq	DB sequences file	DDL Tool
mf	System created Java meta file	Java tools
jar	JAR file	Java tools/FCUBS Installer
rar	RAR compressed file	Java tools/FCUBS Installer
war	Web Archive file	Java tools/FCUBS Installer
ear	EAR file	Jdeveloper/FCUBS Installer
log	System created log files	N/A
rpd	OBIEE repository files	OBIEE
dbc	DB template installation file	Oracle Database
dfb	DB template installation file	Oracle Database
fnc	DB functions	PLSQL Developer
prc	DB procedures	PLSQL Developer

trg	DB Trigger	PLSQL Developer
typ	PLSQL Type file	DDL Tool
vw	DB view file	PLSQL Developer
spc	DB package spec	PLSQL Developer/RAD tool
sql	DB package body	PLSQL Developer/RAD tool
xml	XML file	RAD/Textpad/Jdeveloper/TrAX
xsd	XML schema definition file	RAD/TrAX
sh	Unix shell script	Text pad
xdo	BIP report file	Text pad
adf	GI ascii format definition	Textpad
bat	windows batch file	Textpad
c	C program file	Textpad
cmd	windows batch file	Textpad
ddf	FLEXML - DDF type definition	Textpad
fmt	Advice format	Textpad
frm	Advice format	Textpad
properties	Properties file	Textpad
txt	Text file	Textpad
java	Java source	Textpad/Jdeveloper
js	Java script file	Textpad/Jdeveloper
jsp	JSP file	Textpad/Jdeveloper
wsdl	Web service definition file	TrAX
rtf	BIP RTF report file	Windows word with BIP plug-in

3.8.9 **RAD object naming conventions**

RAD function ID development need to follow the below naming conventions

Table 3.5 – RAD naming convention list

Item	*Prefix*	*Example*
LOV	LOV_	LOV_BRANCH
Data Block	BLK_	BLK_STTM_CUSTOMER
Labels	LBL_	LBL_CUSTOMER_NO
XSD Tags	*Remove hyphen in column name*	CUSTOMERNO
XSD node names	*From block, replace _ with -*	Sttm-Customer
Buttons	BTN_	BTN_EXIT
Screens	CVS_	CVS_MAIN
Tabs	TAB_	TAB_AUXILIARY
Sections	SECn	SEC1
Partitions	Pn	P2
Field Sets	FST_	FST_1

3.9 FLEXCUBE Hand Coded / Manually developed Components

Other than the RAD generated pl/sql source code, FLEXCUBE UBS consists of core processing database packages. These packages are developed manually. These packages not allowed to be modified as a part of extensibility or customizations.

For e.g.

- ACPKS - Performs accounting services
- CYPKS - Performs all currency serices
- ISPKS - Performs Settlement related services
- Wrp_batch - Enables a function ID to be executed from Auto EOD process (AEOD)

4 FLEXCUBE UBS Application Developer Documents

This section describes and links the various Developer documents that are required for FLEXCUBE UBS Application components development.

4.1 Document classifications

The documents are classified as three types

Table 4.1 – Developer documents classification list

Type	*What it contains*	*Documents*
Concepts	This explains the concepts and bring user to development context.	• FCUBS-FD01-01-01-Development Overview Guide • FCUBS-FD02-01-01-RAD Getting Started • FCUBS-FD03-01-01-Extensibility Getting started • FCUBS-FD04-01-01-Interface Getting started • FCUBS-FD05-01-01-Tools-Getting Started • FCUBS-FD06-01-01-Support Getting started • FCUBS-FD07-01-01-Report Getting started • FCUBS-FD08-01-01-Data Model getting started
Procedure	This explains step by step procedure with screen shots	• FCUBS-FD02-02-01-RAD Function ID Development Volume 1

	on how to do a given component development	• FCUBS-FD02-02-02-RAD Function ID Development Volume 2 • FCUBS-FD02-03-01-RAD Web Service Development ▪ FCUBS-FD02-04-01-RAD BIP Report Integration ▪ FCUBS-FD02-04-01-RAD BIP Report Integration ▪ FCUBS-FD02-05-01-RAD Notification Development ▪ FCUBS-FD03-03-01-Extensibility By Example Volume 1 ▪ FCUBS-FD03-03-02-Extensibility By Example Volume 2 ▪ FCUBS-FD04-02-01-Generic Interface Configuration Guide ▪ FCUBS-FD04-03-01-Upload Adapter Development Guide ▪ FCUBS-FD06-02-01-Support By Example ▪ FCUBS-FD07-02-01-BIP Report Development Guide • FCUBS-FD07-03-01-OBIEE repository Development Guide
Reference	These are reference documents provided by tools for all possible features.	▪ FCUBS-FD03-02-01-Extensibility Reference Guide ▪ FCUBS-FD05-02-01-RAD-Reference ▪ FCUBS-FD05-02-02-RAD-Installation and Setup ▪ FCUBS-FD05-03-01-DDL-Reference ▪ FCUBS-FD05-04-01-TrAX-Reference

4.2 Document contents

This section briefs the contents of the developer documents available for FLEXCUBE UBS application development.

- FCUBS-FD01-01-01-Development Overview Guide

  This provides the bird's eye view of FLEXCUBE UBS Development and set the basic foundation for developer including concepts, architecture, framework, tools and global glossary. *This is the document that you are currently reading.*

  - RAD
    - FCUBS-FD02-01-01-RAD Getting Started
      This document gives head start to use RAD tool for FLEXCUBE Application development. It covers the RAD development life cycle and specification needed to develop RAD function IDs.

    - FCUBS-FD02-02-01-RAD Function ID Development Volume 1
      This document explains the step by step procedure to develop the function ID using extensible RAD.

    - FCUBS-FD02-02-02-RAD Function ID Development Volume 2
      This document explains the step by step procedure to develop the function ID using non extensible RAD.

    - FCUBS-FD02-03-01-RAD Web Service Development
      This document explains the step by step procedure to develop Web service. It covers development life cycle, deployment and testing of web service.

    - FCUBS-FD02-04-01-RAD BIP Report Integration
      This document explains the step by step procedure to integrate the BIP developed in BIP server with FLEXCUBE UBS function ID. This integration helps to launch BIP reports from FLEXCUBE UBS URL.

    - FCUBS-FD02-05-01-RAD Notification Development
      This document explains the step by step procedure to create Notifications using pre-developed query web services and deployment and testing.

  - Extensibility
    - FCUBS-FD03-01-01-Extensibility Getting started
      This document gives head start to work on FLEXCUBE UBS extensible framework. It explains the various business areas that extensibility available and concepts behind it.

    - FCUBS-FD03-02-01-Extensibility Reference Guide
      This reference guide provides extensibility framework concepts and features.

- FCUBS-FD03-03-01-Extensibility By Example Volume 1
  This document explains the step by step extensible development with simple examples that includes layout changes, addition of UI elements and functional extensibility using SDE/UDF.

- FCUBS-FD03-03-02-Extensibility By Example Volume 2
  This document explains the step by step extensible development with medium complex example that covers processing and UI changes.

- Interface
  - FCUBS-FD04-01-01-Interface Getting started
    This document explains the various Integration and interface concepts of FLEXCUBE UBS. It covers the overview diagrams to set context.

  - FCUBS-FD04-02-01-Generic Interface Configuration Guide
    This document explains step by step procedure to define GI outgoing and incoming interface.

  - FCUBS-FD04-03-01-Upload Adapter Development Guide
    This document explains step by step procedure to create spreadsheet based FCUBS upload adapters that are used for data migration into FLEXCUBE UBS.

- Tools
  - FCUBS-FD05-01-01-Tools-Getting Started
    This document gives head start for development tools available .

  - FCUBS-FD05-02-01-RAD-Reference
    This document provides reference information of extensible RAD that includes all features available and concepts

  - FCUBS-FD05-02-02-RAD-Installation and Setup
    This document provides the installation and setup steps for extensible RAD

  - FCUBS-FD05-03-01-DDL-Reference
    This document provides the reference information of DDL tool features and its usage.

    Note: *This tool in internal to Oracle Financial Software Services and this document may not be available externally.*

  - FCUBS-FD05-04-01-TrAX-Reference
    This document provides the reference information of TrAX tool features and its usage.
    Note: *This tool in internal to Oracle Financial Software Services and this document may not be available externally.*

- Support
  - FCUBS-FD06-01-01-Support Getting started
    This document provides the possible FLEXCUBE UBS software issues and various tools available for support.

  - FCUBS-FD06-02-01-Support By Example
    This document provides uses cases that explain the FLEXCUBE UBS software support that covers issue analyze/fix.

- Reports
  - FCUBS-FD07-01-01-Report Getting started
    This document gives head starts on reports development in FLEXCUBE UBS using BIP or OBIEE meta data repository.

  - FCUBS-FD07-02-01-BIP Report Development Guide
    This document explains the step by step procedure to develop the BIP report.

  - FCUBS-FD07-03-01-OBIEE repository Development Guide
    This document explains the step by step procedure to develop OBIEE Meta data repository.

- Data Model
  - FCUBS-FD08-01-01-Data Model getting started
    This document helps to start using FLEXCUBE UBS data model artifacts.

- Dashboard
  - FCUBS-FD09-01-01-Dashboard_Developement
    This document describes the steps to develop FLEXCUBE UBS Dashboards.

5 Developer Glossary

This section provides the developer glossaries that are applicable in all developer documents.

Acronym	*Meaning*
AUDF	Ascii User Defined Field
Back-end	Represents the Database layer
BIP	Business Intelligence Publisher
BLK	Block (used in RAD screen development)
CI	Configurable Item
CVS	Canvas
DDL	Data Definition Language
DOM	Document Object Model
DSN	Data Source Name (Microsoft ODBC)
EAR	Enterprise ARchive file
EJB	Enterprise Java Bean
EMS	Electronic Media System
EOD	End Of Day
ER	Entity Relationship
FC BRN	FLEXCUBE Branch
FCJ	FLEXCUBE Java
Front-end	Represents the client layer(browser)
FS-FS	Full Screen - Full Screen (Web service pattern)
FST	Field Set (used in RAD screen development)
FTP	File Transfer Protocol
GI	Generic Interface
GW_WS	Gateway Web Service
IC	Interest and Charges
ICEOD	Interest and Charges End of Day
IDE	Integrated Development Environment
IE	Microsoft Internet Explorer
IMPL	Implementation files (used in web services)
INC	File extension used to represent static data (Insert statements)
IO-FS	Input Only - Full Screen (Web service pattern)
IO-PK	Input Only - Primark Key (Web service pattern)
JDBC	Java Data Base Connectivity
JEE	Java Enterprise Edition
JMS	Java Messaging Standard
JS	Java Script file

LBL	Label (used in RAD screen development)
LOV	List Of Value (used in RAD screen development)
MDB	Message Driven Bean
MICON	FLEXCUBE UBS Management and Integration Console
MSG	Message
NQS	Network Queuing System
OBIEE	Oracle Business Intelligence Enterprise Edition
OLTP	On Line Transaction Processing
PK	Primary Key of Database Table
PK_Cols	Primark Key columns names
PK_Types	Primark Key columns types
RAD	Rapid Application Development Tool
RPD	OBIEE Repository(meta data) file
RTF	Rich Text Format
SDE	System Data Element
SEC	Section (used in RAD screen development)
SEQ	Oracle database SEQuence
SMS	Security Management System
SOAP	Simple Object Access Protocol
SPC	Oracle database package SPeCification
SQL	Oracle database package body
SWIFT	Society for Worldwide International Fund Transfer
SYS	System Java script file
TIX	Tilda separated in XML format
TNS	Oracle TNS entries
TrAX	Tracking and Analyzing xsd's Tool
TS	Tilda Separated
UBS	FLEXCUBE Universal Banking Solution
UDF	User Defined Field
UI	User Interface
UIXML	User Interface XML (runtime file)
WAR	Web Archive file
WS	Web Service
WSC	Web Service Custodian
WSDL	Web Service Description Language
XDO	Extensible Data Object
XHTTP	XML HTTP format
XML	Extensible Markup Language
XSD	XML Schema Definition

6 List of Figures

7 List of Tables

Development Overview Guide
May 2012
1.0

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com/ financial_services/