Oracle® Banking Enterprise Limits and Collateral Management

Development Workbench - Notifications

Release 14.8.0.0.0
G32526-01
April 2025

Oracle Banking Enterprise Limits and Collateral Management Development Workbench - Notifications, Release 14.8.0.0.0

G32526-01

Copyright © 2007, 2025, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1
Preface

1.1 Purpose

This guide is designed to help acquaint you with the Oracle Banking Enterprise Limits and Collateral Management (ELCM) application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

User can further obtain information specific to a particular field by placing the cursor on the relevant field and pressing <F1> on the keyboard.

1.2 Audience

This guide is intended for the following User/User Roles:

Table 1-1 Audience

Role	Function
Back office data entry clerk	Input functions for funds
Back office managers/officers	Authorization functions
Product Managers	Product definition and authorization
End of day operators	Processing during end of day / beginning of day

1.3 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.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

1.4 Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

1.5 Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1.6 Basic Actions

Table 1-2 Basic Actions

Action	Description
Approve	Used to approve the initiated report. This button is displayed, once the user click **Authorize**.
Audit	Used to view the maker details, checker details, and report status.
Authorize	Used to authorize the report created. A maker of the screen is not allowed to authorize the report. Only a checker can authorize a report, created by a maker.
Close	Used to close a record. This action is available only when a record is created.
Confirm	Used to confirm the performed action.
Cancel	Used to cancel the performed action.
Compare	Used to view the comparison through the field values of old record and the current record. This button is displayed in the widget, once the user click **Authorize**.
Collapse All	Used to hide the details in the sections. This button is displayed, once the user click **Compare**.
Expand All	Used to expand and view all the details in the sections. This button is displayed, once the user click **Compare**.
New	Used to add a new record. When the user click **New**, the system displays a new record enabling to specify the required data.

Table 1-2 (Cont.) Basic Actions

Action	Description
OK	Used to confirm the details in the screen.
Save	Used to save the details entered or selected in the screen.
View	Used to view the report details in a particular modification stage. This button is displayed in the widget, once the user click **Authorize**.
View Difference only	Used to view a comparison through the field element values of old record and the current record, which has undergone changes. This button is displayed, once the user click **Compare**.
Unlock	Used to update the details of an existing record. System displays an existing record in editable mode.

1.7 Related Documents

For more information refer to the Oracle Banking manuals on:

- Development of Launch Forms and Others Screens
- Enterprise Collaterals User Guide
- Enterprise Limits and Collaterals Common User Guide

1.8 Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1.9 Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

1.10 Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1-3 Acronyms and Abbreviations

Acronyms	Abbreviations
CIF	Customer Information File

Table 1-3 (Cont.) Acronyms and Abbreviations

Acronyms	Abbreviations
CASA	Current Account and Savings Account
DDA	System that holds the CASA account and balances
ELCM	Enterprise Limits and Collateral Management
ECA	External Credit Approval
FCUBS	Oracle FLEXCUBE Universal Banking Solution
GW	Gateway
HTTP	Hyper Text Transfer Protocol
ID	Identification Number
Mark EOTI	Mark End of Transaction Input
Mark TI	Mark Transaction Input
OFSAA	Oracle Financial Services Analytical Applications
ORMD	Oracle Revenue and Billing Management
PK	Primary Key
RDBMS	Relational Data Base Management System
SMS	Security Services
UI	User Interface
VD	Value Date
XML	Extensible Mark-up Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations

1.11 Symbols and Icons

The lists of symbols, buttons and shortcut key that are used in the application to perform various tasks are covered in this topic.

Table 1-4 Symbols and Icons

Icons	Function
🔍	Perform search
⌟⌞ ⌝⌜	Minimize
▶	Navigate to the next record
◀	Navigate to the previous record
○	Toggle OFF
●	Toggle ON
✕	Delete
+	Click this icon to add a new row.

Table 1-4 (Cont.) Symbols and Icons

Icons	Function
—	Click this icon to delete an existing row.
☰	List view
⛶	Maximize
|<	Navigate to the first record
>|	Navigate to the last record
🔍	Advance search
✎	Search record
💾	Save the record
↺	Reset the record
📋	Clear the record

Table 1-5 Symbols and Icons - Audit Details

Icons	Function
👤	A user
🏛	Branch details
📅	Date and Time

1.12 Prerequisite

Specify the **User ID** and **Password**, and login to **Home** screen.

2 Introduction

3 Notification – Getting started

This topic provides an overview of the Notification framework in FLEXCUBE UBS.

Notification

Notification framework in FLEXCUBE UBS is used to communicate the business event that happened in FLEXCUBE UBS to external systems. Depending upon the event, the XML message is pushed to the external system’s asynchronous Queues for their consumption.

Notification Trigger

Notification Triggers is developed to recognize the event and then invoke the notification process. This trigger is developed using Development Workbench.

4 Notification Development

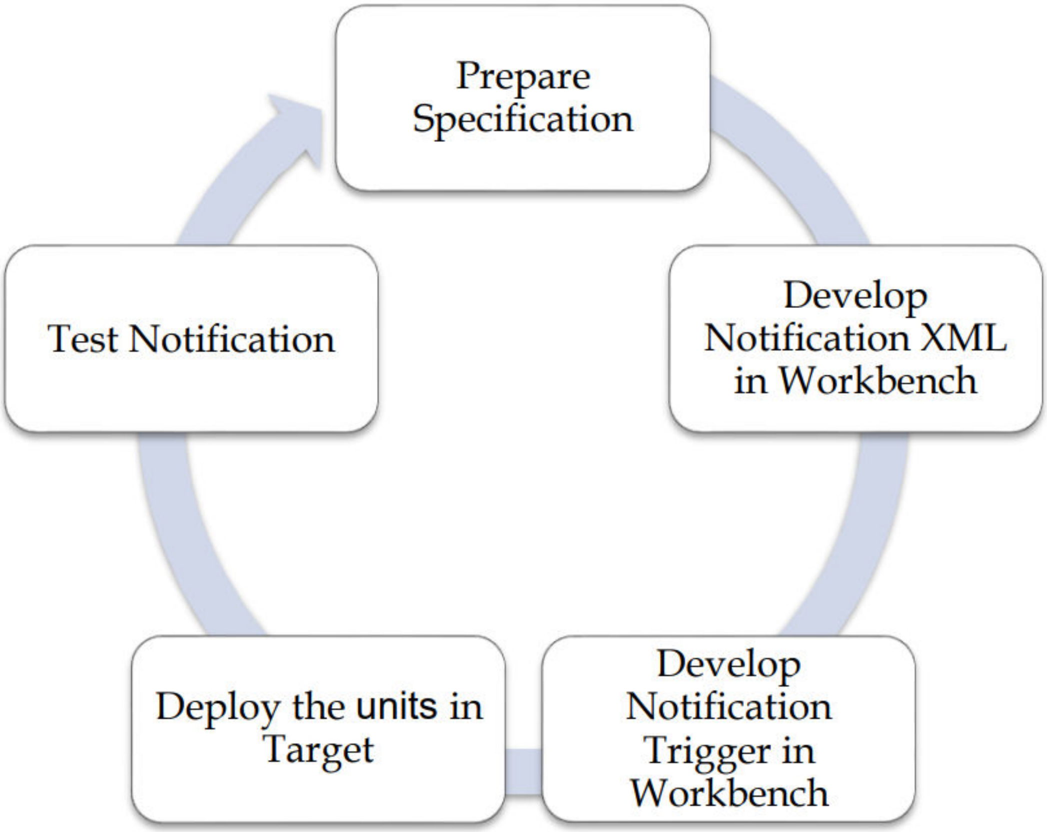
This topic provides an overview of the Notification Development process.

Prerequisite for Notification development and testing

Following are the prerequisite for notification development:

- Target FLEXCUBE Environment with Notification framework installed.
- Development Workbench link mapped to the FLEXCUBE environment.
- Required Query Web services developed and tested.

Figure 4-1 Architecture- Notification Development


Prepare Specification
Develop Notification XML in Workbench
Develop Notification Trigger in Workbench
Deploy the units in Target
Test Notification

Notification specification

Identify the notification requirement as below:

- What is the Notification function ID name for RAD XML (the Third character should be N)?

- What is the Notification code?
- What is the Base table in FLEXCUBE UBS that triggers the notification?
  - What operation at base table triggers (insert/update/delete)?
  - What is the where clause for the filter?
- What is the query Web service to be used?
  - What is the operation?
  - What are the tags required?

For Example:

- Notification function ID name – STNCUMOD
- Notification code – NOTIF_CA_CUSTACC_MOD
- Base table - STTM_CUST_ACCOUNT
  - Operation – DELETE
  - Filter – Account class type in (S,U)
- Web service to be used – FCUBSAccService
  - Operation – QueryCustAcc
  - Request node – Cust-Account-IO

Notification XML development

Notification RAD XML development creates the following files:

- RAD XML
- SPC
- SQL
- Static Data

Notification Process

There will be one trigger for the base table of notification and in case of multiple notifications sharing the same base table, there will be no new triggers created. Instead, the same trigger created on the base table will be reused. This approach reduces the number of triggers being used for notifications.

Development process in Development Workbench

The notification development process in Workbench is split into two steps:

1. Notification Triggers
2. Notification Filter Procedure

The first step is to create notification triggers for base tables. The trigger generated from Workbench will be inserting key details into a static notification log table. The following details will be captured:

- Trigger code: A unique value for a notification trigger.
- Base Table: The base table on which, the trigger is built.
- When Clause: A simple when clause for the notification trigger.

The second step is to capture the details of notifications and generate the notification filter procedure. The following details are captured:

- Notification code: A unique value to identify a notification.
- Description: Meaningful description of the notification.
- Gateway Service

5

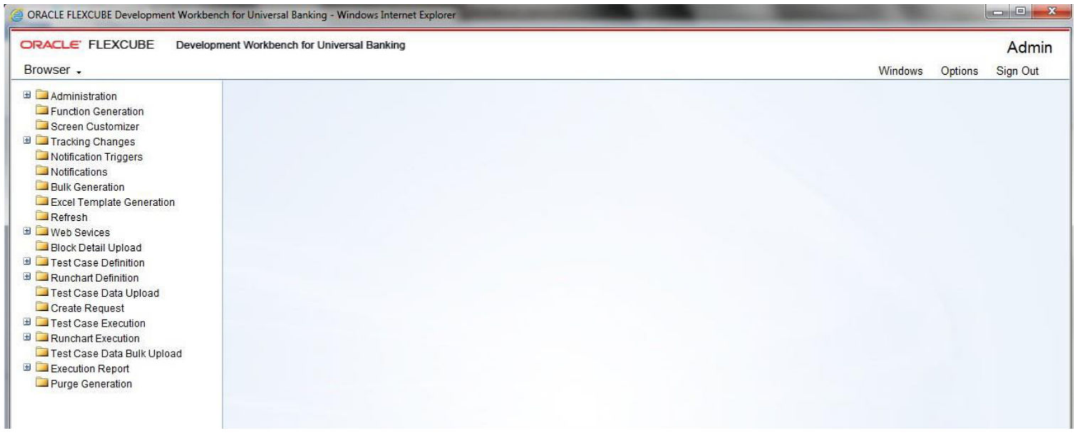
Generate Notification Trigger

This topic provides systematic instructions to generate new or modify the existing Notification Trigger.

1. Log in to the Development Workbench.

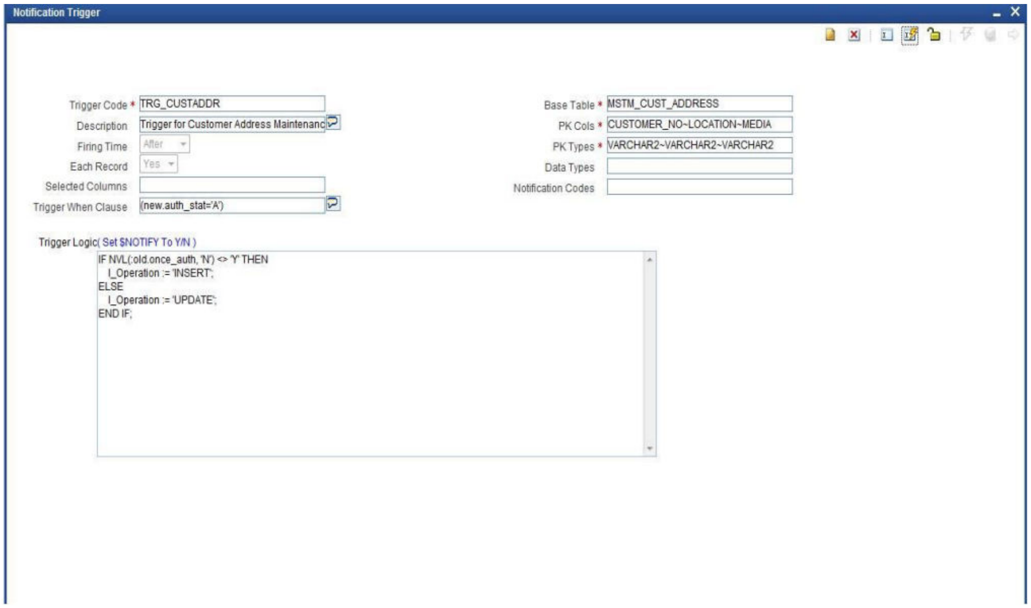
   Development Workbench For Universal Banking screen displays.

   Figure 5-1 Development Workbench For Universal Banking

2. Click on the **Notification Triggers** option under the **Browser** menu.
3. To add a new Notification Trigger, follow the steps given below:
   - Specify the details in the **Notification Trigger- New** screen and save the Notification Trigger.

     Figure 5-2 Notification Trigger- New

     Notification Trigger

     Trigger Code * TRG_CUSTADDR

     Description Trigger for Customer Address Maintenanc

     Firing Time After

     Each Record Yes

     Selected Columns

     Trigger When Clause (new.auth_stat='A')

     Base Table * MSTM_CUST_ADDRESS

     PK Cols * CUSTOMER_NO~LOCATION~MEDIA

     PK Types * VARCHAR2~VARCHAR2~VARCHAR2

     Data Types

     Notification Codes

     Trigger Logic(Set $NOTIFY To Y/N)

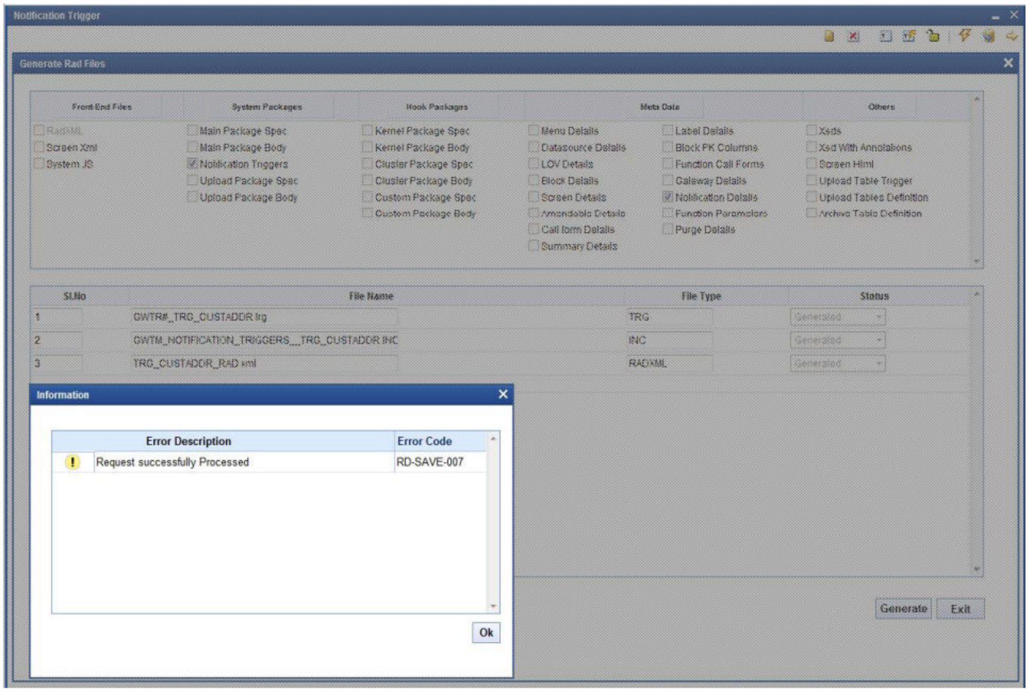
```
IF NVL(:old.once_auth, 'N') <> 'Y' THEN
   l_Operation := 'INSERT';
ELSE
   l_Operation := 'UPDATE';
END IF;
```

Table 5-1 Notification Trigger- New

Field	Description
Trigger Code	A unique value for a notification trigger. The naming conversion should start with **TRG_XXXX**. This is a mandatory field. This attribute signifies the trigger code created as part of the trigger creation step in OTD. Each notification will be linked to a trigger code.
Description	This is an information field. Specify a meaningful description of Trigger.
Firing Time	Specify when the trigger needs to be fired. The user can create only **Before** and **After** triggers for tables. (**INSTEAD OF** triggers are only available for views; typically they are used to implement view updates.) (**After**/**Before**).
Each Record	Specify for each row required or not. If **For Each Row** option is specified, the trigger is row-level; otherwise, the trigger is statement-level. (**Yes**/**No**)
Base Table	The base table on which, the trigger is built. This is a mandatory field. Select a valid table from available LOV next to the field.
Pk Cols	Enter Primary key fields of table in tilde (~) separated format. This is a mandatory field.
Pk Types	Enter the Primary key type of the corresponding primary key field. This is a mandatory field.
Selected Columns and Data Types	Defunct
Trigger When Clause	A simple when clause for the notification trigger. A trigger restriction can be specified in the WHEN clause, enclosed by parentheses. The trigger restriction is a SQL condition that must be satisfied for Oracle to fire the trigger. This condition cannot contain sub queries. Without the WHEN clause, the trigger is fired for each row.
Notification Codes	If the trigger is associated with a specific notification code, then the particular notification code has to be provided in the field. If the trigger is shared across many Notifications, the field can be left empty.

An **Information** window displays with confirming the message of successful generation of new Notification Trigger.

Figure 5-3   Information



Front End Files	System Packages	Hook Packages	Meta Data		Others
□ RadXML	□ Main Package Spec	□ Kernel Package Spec	□ Menu Details	□ Label Details	□ Xsds
□ Screen Xml	□ Main Package Body	□ Kernel Package Body	□ Datasource Details	□ Block PK Columns	□ Xsd With Annotations
□ System JS	☑ Notification Triggers	□ Cluster Package Spec	□ LOV Details	□ Function Call Forms	□ Screen Html
	□ Upload Package Spec	□ Cluster Package Body	□ Block Details	□ Gateway Details	□ Upload Table Trigger
	□ Upload Package Body	□ Custom Package Spec	□ Screen Details	☑ Notification Details	□ Upload Tables Definition
		□ Custom Package Body	□ Amendable Details	□ Function Parameters	□ Archive Table Definition
			□ Call form Details	□ Purge Details	
			□ Summary Details		

Sl.No	File Name	File Type	Status
1	GWTR#_TRG_CUSTADDR.trg	TRG	Generated
2	GWTM_NOTIFICATION_TRIGGERS__TRG_CUSTADDR.INC	INC	Generated
3	TRG_CUSTADDR_RAD.xml	RADXML	Generated

	Error Description	Error Code
!	Request successfully Processed	RD-SAVE-007

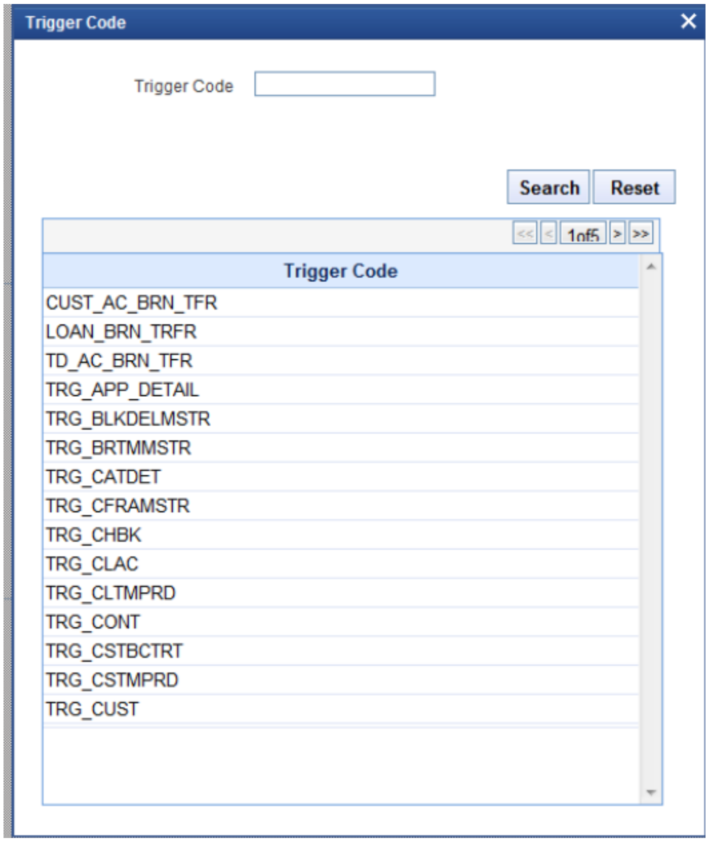
On successful save, Notification Trigger generates two files (**gwtr#_<trg-code>.trg** and **GWTM_NOTIFICATION_TRIGGERS__<trg-code>.INC**) that user needs to compile them in FLEXCUBE schema.

4. To modify the existing Notification Trigger, follow the steps given below:

   a. In **Notification Trigger** screen, click on the browse icon at **Trigger Code** option.

      Trigger Code window displays.

Figure 5-4 Trigger Code



Trigger Code

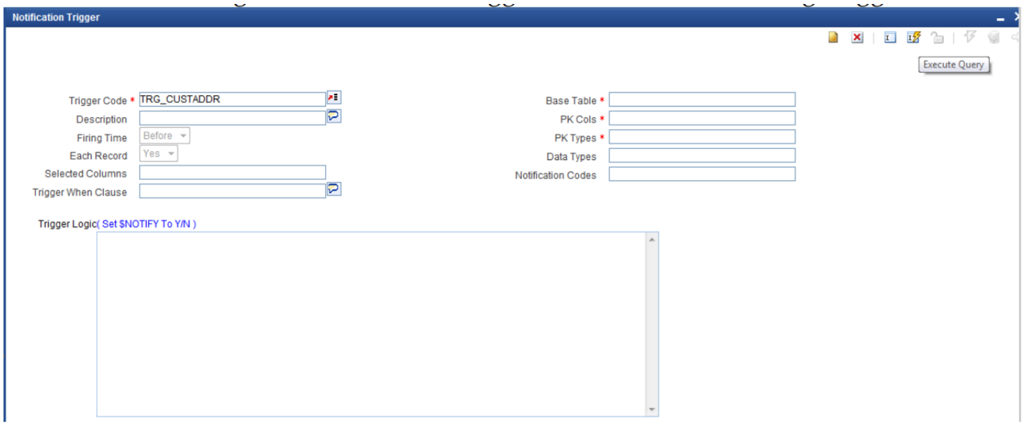
Trigger Code
CUST_AC_BRN_TFR
LOAN_BRN_TRFR
TD_AC_BRN_TFR
TRG_APP_DETAIL
TRG_BLKDELMSTR
TRG_BRTMMSTR
TRG_CATDET
TRG_CFRAMSTR
TRG_CHBK
TRG_CLAC
TRG_CLTMPRD
TRG_CONT
TRG_CSTBCTRT
TRG_CSTMPRD
TRG_CUST

b. Enter the trigger to be modified in the **Trigger code** field and click on **Search** button.

c. Select the **Trigger Code** from the list.

   Notification Trigger screen displays with the selected **Trigger code**.

Figure 5-5 Notification Trigger



Trigger Code *	TRG_CUSTADDR	Base Table *	
Description		PK Cols *	
Firing Time	Before	PK Types *	
Each Record	Yes	Data Types	
Selected Columns		Notification Codes	
Trigger When Clause			

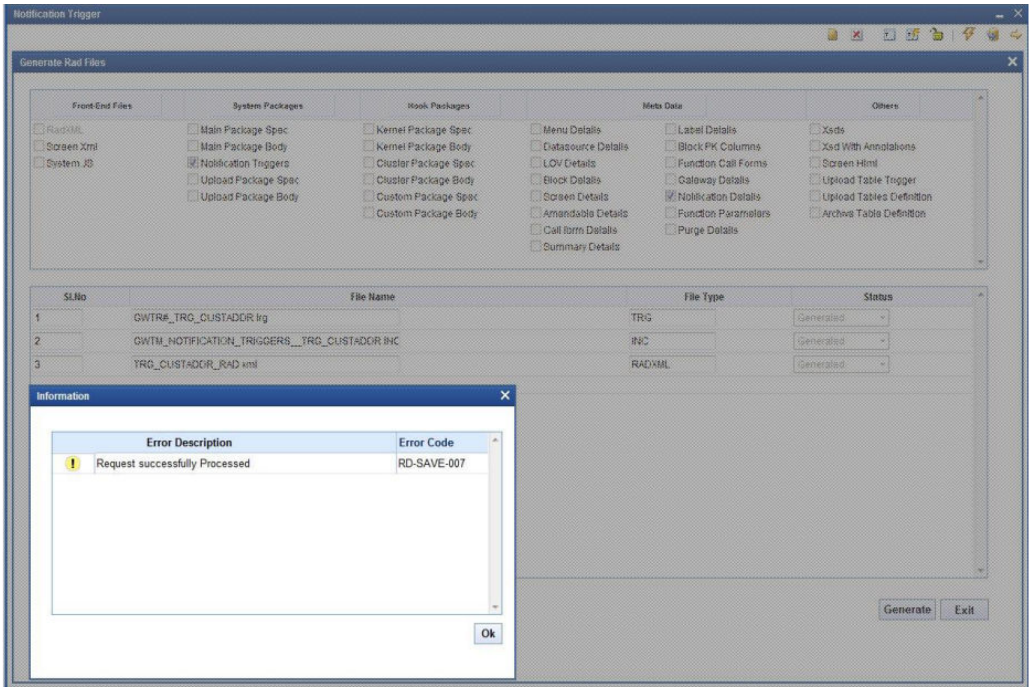
Trigger Logic(Set $NOTIFY To Y/N)

d. Specify the new details to modify the existing one and save the Notification Trigger.

Figure 5-6 Notification Trigger- Modify

An **Information** window displays with confirming the message of successful modification of the Notification Trigger.

Figure 5-7 Information



Sl.No	File Name	File Type	Status
1	GWTR#_TRG_CUSTADDR.trg	TRG	Generated
2	GWTM_NOTIFICATION_TRIGGERS__TRG_CUSTADDR.INC	INC	Generated
3	TRG_CUSTADDR_RAD.xml	RADXML	Generated

	Error Description	Error Code
!	Request successfully Processed	RD-SAVE-007

6 Deploy Notification

This topic provides systematic instructions to deploy Notification.

1. For Notification - Workbench related deployment, compile the following files in Target FLEXCUBE UBS Database schema:
   - Notification Main Package generated from ODT
   - Hook Packages
   - GWTM_NOTIFICATION_TAG_MAP___<Notification Function ID>_.INC
   - GWTM_NOTIFICATIONS_MASTER___<Notification Function ID>_.INC
2. For Notification Trigger deployment, compile the following files in Target FLEXCUBE UBS Database schema:
   - GWTM_NOTIFICATION_TRIGGERS__TRIG_CONTRACT.INC
   - GWTR#_TRIG_CONTRACT.TRG

7

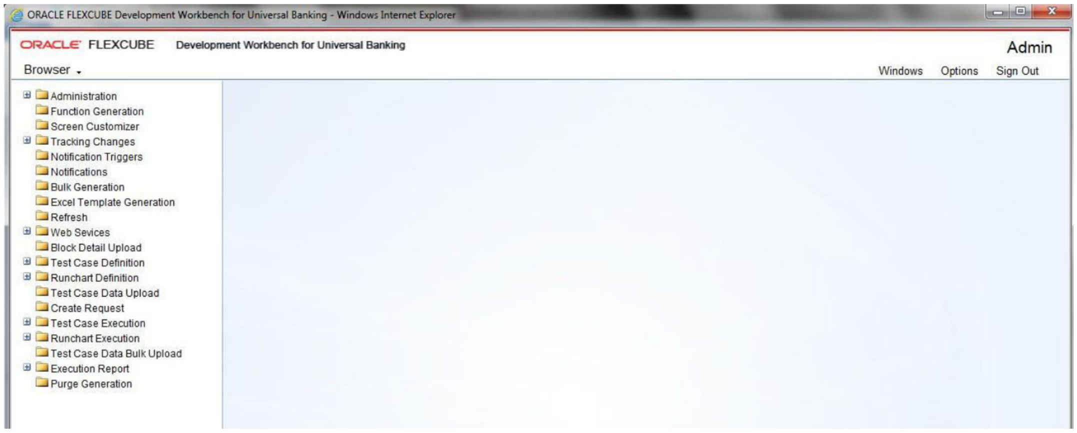
Create Notifications

This topic provides systematic instructions to generate new or modify the existing Notification.

1. Log in to the Development Workbench.

   Development Workbench For Universal Banking screen displays.

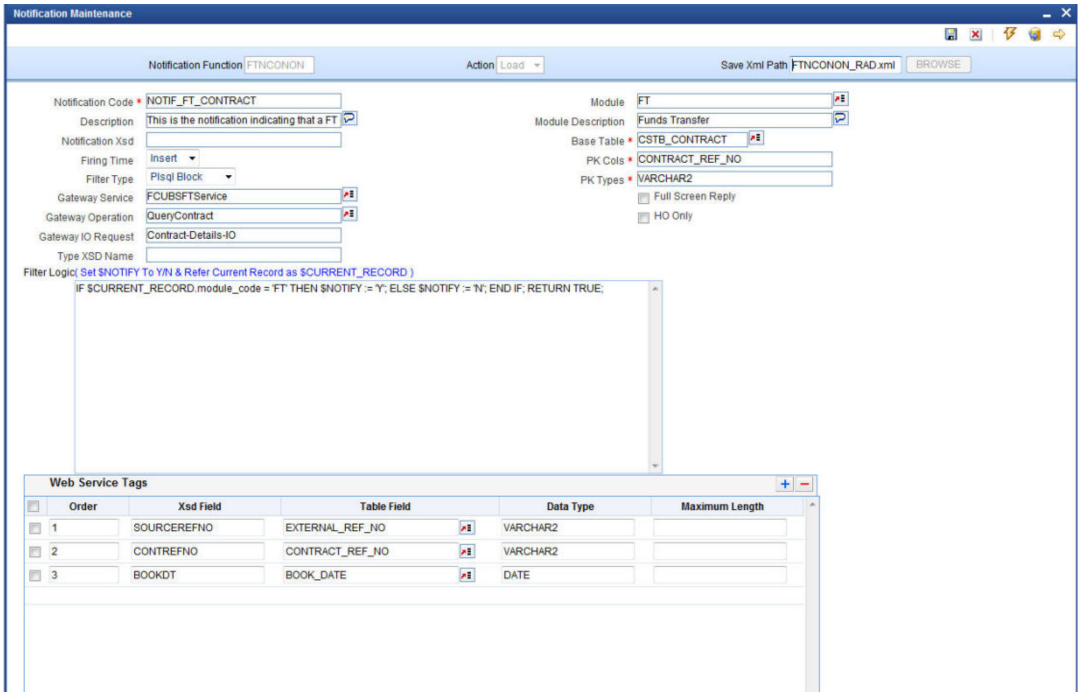
   Figure 7-1 Development Workbench For Universal Banking

2. Click on the **Notifications** option under the **Browser** menu.

   Notification Maintenance screen displays.

   Figure 7-2 Notification Maintenance

Notification Function	FTNCONON
Action	Load
Save Xml Path	FTNCONON_RAD.xml
Notification Code *	NOTIF_FT_CONTRACT
Description	This is the notification indicating that a FT
Notification Xsd	
Firing Time	Insert
Filter Type	Plsql Block
Gateway Service	FCUBSFTService
Gateway Operation	QueryContract
Gateway IO Request	Contract-Details-IO
Type XSD Name	
Module	FT
Module Description	Funds Transfer
Base Table *	CSTB_CONTRACT
PK Cols *	CONTRACT_REF_NO
PK Types *	VARCHAR2
□ Full Screen Reply	
□ HO Only	

Filter Logic(Set $NOTIFY To Y/N & Refer Current Record as $CURRENT_RECORD)

```
IF $CURRENT_RECORD.module_code = 'FT' THEN $NOTIFY := 'Y'; ELSE $NOTIFY := 'N'; END IF; RETURN TRUE;
```

□	Order	Xsd Field	Table Field	Data Type	Maximum Length
□	1	SOURCEREFNO	EXTERNAL_REF_NO	VARCHAR2	
□	2	CONTREFNO	CONTRACT_REF_NO	VARCHAR2	
□	3	BOOKDT	BOOK_DATE	DATE	

Notification Maintenance screen is used to create a new notification or modify an existing notification, here notification information gets captured for notification codes and the user needs to save notification details into XML.

3. To create a new Notification, follow the steps given below:
    a. Specify the details in the **Notification Maintenance** screen and save it.

Table 7-1 Notification Maintenance- Field Description

Field	Description
Action	Select either **New** or **Load** action. The **New** option is to create a new notification and the **Load** option is used to modify the existing one.
Save XML Path	Specify the path to save notification XML. This would be considered only if the Save Mode is Client and Work Directory specified as **$CURRENT_DIRECTORY**.
Notification Function	Specify the notification function-id name. **Note:** Naming Conventions- Notification Function must have maximum 8 characters and the 3rd letter must be **N**. For Example: **FTNCONON**
Notification Code	Enter the notification code to which we need to capture values. This is a mandatory field. **Note:** Recommended Convention for Notification Codes- NOTIF_<Module Code>_<Description> For Example: **NOTIF_LD_CONTRACT**, This is the notification indicating that an LD contract has been created/modified.
Description	Information field. A meaningful description of the Notification has to be provided in the field.
Module	This attribute signifies the module on which the notification is based.
Module Description	Information field. Module Description that would be defaulted from Module LOV.

Table 7-1 (Cont.) Notification Maintenance- Field Description

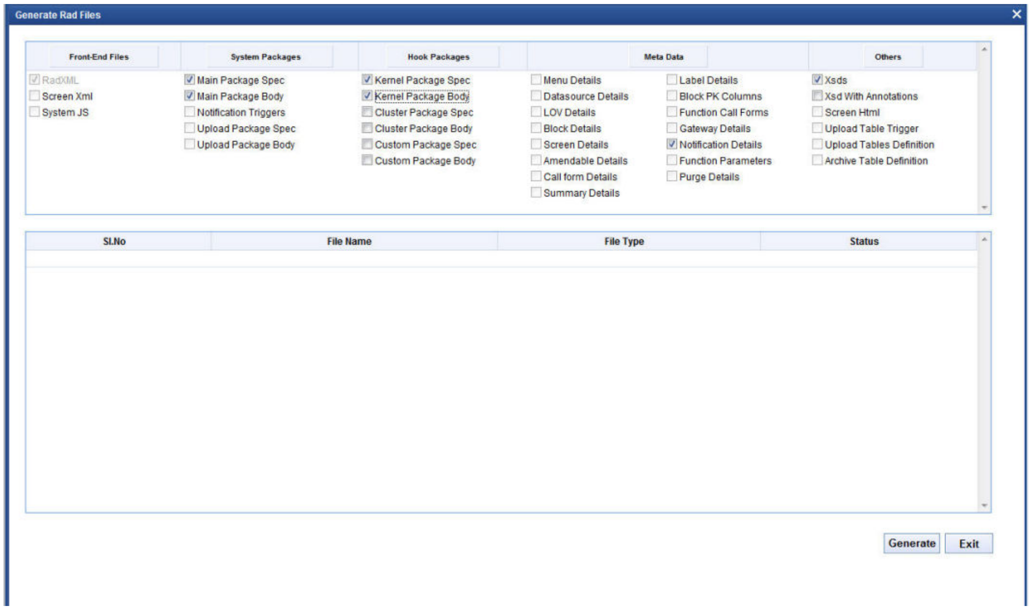
Field	Description
Notification XSD	Notification XSD name will have to be provided in the corresponding Field. **Note:** The naming convention to be followed while naming Notification XSD is as follows- [Module Name] – [Notification Description] – Notif.xsd For Example: **FT-Contract-Notif.xsd**, Notification XSD has to be provided only if no Gateway Web Service Query Operation is configured to the Notification.
Base Table	Select the base table on which trigger needs to be applied.
Firing Time	Indicates the Operation on the base Table for which Notifications have to be sent. Options available are **Insert**, **Update**, or both.
Filter Type	This attribute can take the following values. i. **Where Clause** ii. **Plsql Block**
Pk Cols	Enter Primary key columns of the Base Table.
Pk Types	Enter Primary key field Data Types. Provide details of Gateway Service, Operation, Type XSD Name, and Full-Screen Reply if a Query Web Service has to be mapped to the Notification.
Gateway Operation	The gateway operation name to execute the query for the mentioned Service.
Gateway Service	The gateway service is to be used to get the full-screen response.
Gateway IO Request	The gateway IO request node to be used in querying operations.
Type XSD Name	This field has to be entered if Notification is mapped to a Service and Request. The name of the Master Type XSD for the service and operation has to be provided here. This can be found in the included portion of the Request Msg XSD of particular Service-Operation. For Example: **LC-Contract-Types.xsd**
Full screen Reply	This attribute decides whether a full screen or primary key notification response is to be sent. This is applicable only if gateway Service details are provided.
HO only	This attribute is used to send notifications only from the head office.

Table 7-1 (Cont.) Notification Maintenance- Field Description

Field	Description
Filter Logic	The filter logic decides whether the notification needs to be sent or not. This can be a simple where-clause on the base table or a complex pl/sql block.
Web service Tags	The columns selected from the base table as part of web service tags will be used to send the full-screen notification response. These tags define the elements of Notification XML when no Query service is mapped to it.

Generate Rad Files screen displays.

Figure 7-3 Generate Rad Files



Front-End Files
☑ RadXML
☐ Screen Xml
☐ System JS

System Packages
☑ Main Package Spec
☑ Main Package Body
☐ Notification Triggers
☐ Upload Package Spec
☐ Upload Package Body

Hook Packages
☑ Kernel Package Spec
☑ Kernel Package Body
☐ Cluster Package Spec
☐ Cluster Package Body
☐ Custom Package Spec
☐ Custom Package Body

Meta Data
☐ Menu Details
☐ Datasource Details
☐ LOV Details
☐ Block Details
☐ Screen Details
☐ Amendable Details
☐ Call form Details
☐ Summary Details
☐ Label Details
☐ Block PK Columns
☐ Function Call Forms
☐ Gateway Details
☑ Notification Details
☐ Function Parameters
☐ Purge Details

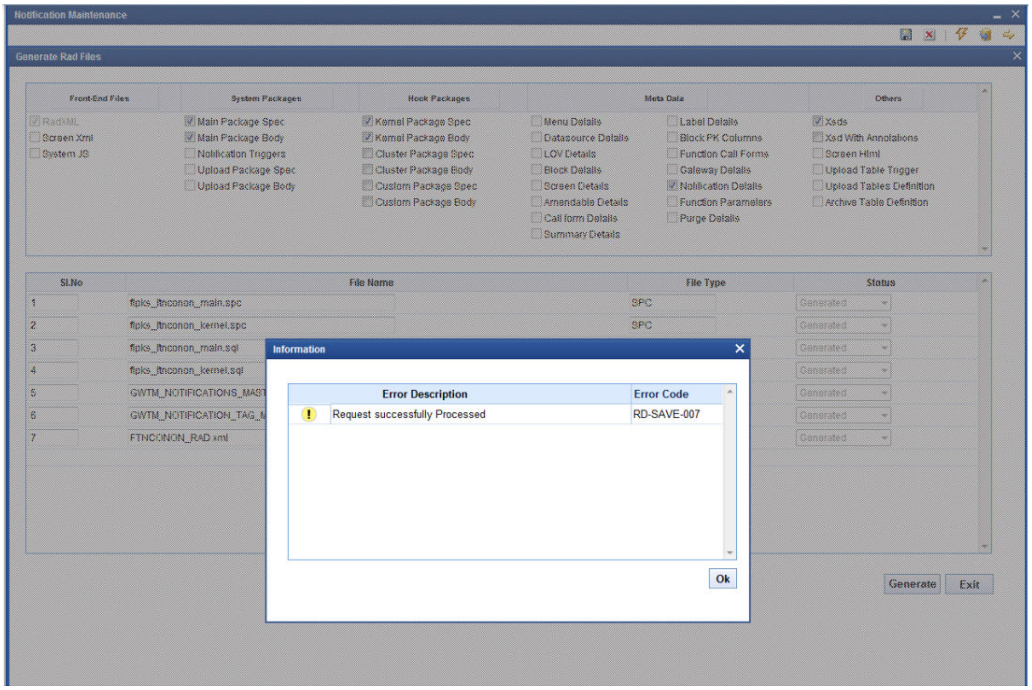
Others
☑ Xsds
☐ Xsd With Annotations
☐ Screen Html
☐ Upload Table Trigger
☐ Upload Tables Definition
☐ Archive Table Definition

Sl.No	File Name	File Type	Status

b. Click on the **Generate** button.

An **Information** window displays confirming the successful generation of Notification.

Figure 7-4   Information



Notification Maintenance

Generate Rad Files

Front-End Files	System Packages	Hook Packages	Meta Data		Others
☑ RadXML	☑ Main Package Spec	☑ Kernel Package Spec	□ Menu Details	□ Label Details	☑ Xsds
□ Screen Xml	☑ Main Package Body	☑ Kernel Package Body	□ Datasource Details	□ Block PK Columns	□ Xsd With Annotations
□ System JS	□ Notification Triggers	□ Cluster Package Spec	□ LOV Details	□ Function Call Forms	□ Screen Html
	□ Upload Package Spec	□ Cluster Package Body	□ Block Details	□ Gateway Details	□ Upload Table Trigger
	□ Upload Package Body	□ Custom Package Spec	□ Screen Details	☑ Notification Details	□ Upload Tables Definition
		□ Custom Package Body	□ Amendable Details	□ Function Parameters	□ Archive Table Definition
			□ Call form Details	□ Purge Details	
			□ Summary Details		

Sl.No	File Name	File Type	Status
1	fipks_ftnconon_main.spc	SPC	Generated
2	fipks_ftnconon_kernel.spc	SPC	Generated
3	fipks_ftnconon_main.sql		Generated
4	fipks_ftnconon_kernel.sql		Generated
5	GWTM_NOTIFICATIONS_MAST		Generated
6	GWTM_NOTIFICATION_TAG_M		Generated
7	FTNCONON_RAD.xml		Generated

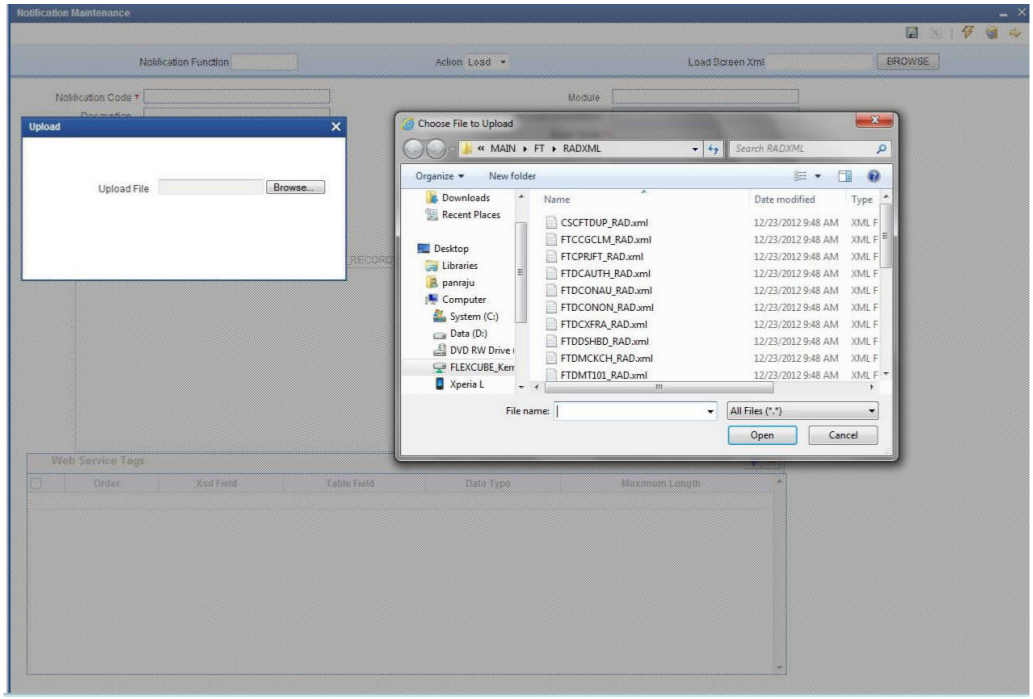
Information

	Error Description	Error Code
!	Request successfully Processed	RD-SAVE-007

4. To modify the existing Notification RADXML, follow the steps given below:
   a. In **Notification Maintenance** screen, select the **Action** field as **Load**.
   b. Click on the **BROWSE** button at **Load Screen XML** field.

      Choose File to Upload folder window displays.

Figure 7-5   Choose File to Upload



Notification Maintenance

Notification Function

Action Load

Load Screen Xml

Notification Code *

Module

Upload

Upload File

Choose File to Upload

« MAIN ▸ FT ▸ RADXML

Name	Date modified	Type
CSCFTDUP_RAD.xml	12/23/2012 9:48 AM	XML F
FTCCGCLM_RAD.xml	12/23/2012 9:48 AM	XML F
FTCPRJFT_RAD.xml	12/23/2012 9:48 AM	XML F
FTDCAUTH_RAD.xml	12/23/2012 9:48 AM	XML F
FTDCONAU_RAD.xml	12/23/2012 9:48 AM	XML F
FTDCONON_RAD.xml	12/23/2012 9:48 AM	XML F
FTDCXFRA_RAD.xml	12/23/2012 9:48 AM	XML F
FTDDSHBD_RAD.xml	12/23/2012 9:48 AM	XML F
FTDMCKCH_RAD.xml	12/23/2012 9:48 AM	XML F
FTDMT101_RAD.xml	12/23/2012 9:48 AM	XML F

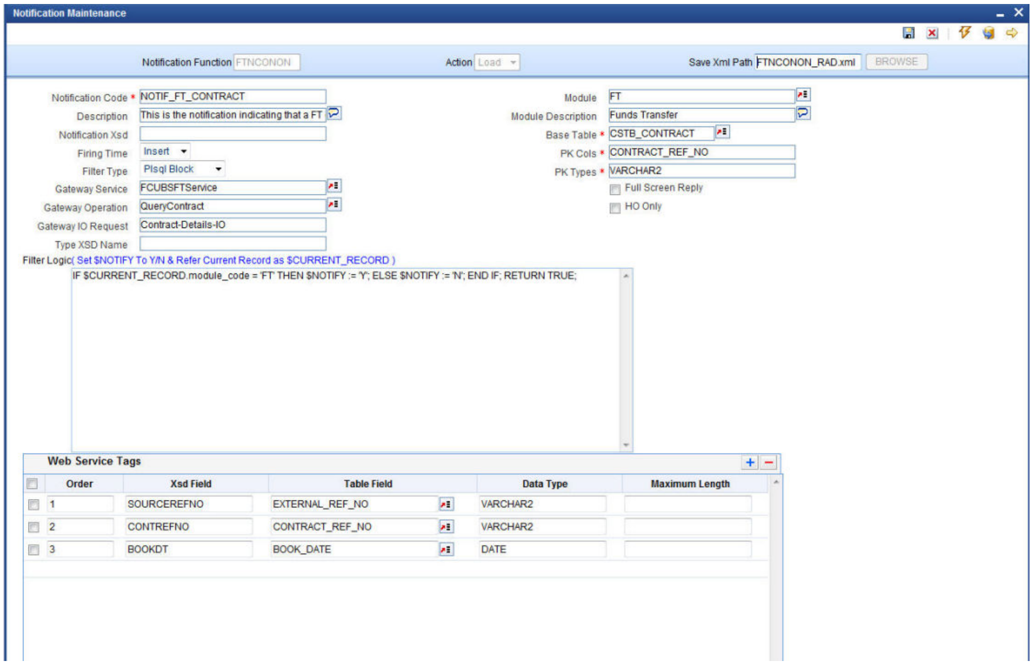
File name:

All Files (*.*)

c. Select the RADXML file to be modified and click on the **Open** button.

   Notification Maintenance screen displays with selected RADXML file details.

Figure 7-6 Notification Maintenance



Notification Maintenance

Notification Function: FTNCONON
Action: Load
Save Xml Path: FTNCONON_RAD.xml

Notification Code *: NOTIF_FT_CONTRACT
Description: This is the notification indicating that a FT
Notification Xsd:
Firing Time: Insert
Filter Type: Plsql Block
Gateway Service: FCUBSFTService
Gateway Operation: QueryContract
Gateway IO Request: Contract-Details-IO
Type XSD Name:

Module: FT
Module Description: Funds Transfer
Base Table *: CSTB_CONTRACT
PK Cols *: CONTRACT_REF_NO
PK Types *: VARCHAR2
□ Full Screen Reply
□ HO Only

Filter Logic(Set $NOTIFY To Y/N & Refer Current Record as $CURRENT_RECORD)

```
IF $CURRENT_RECORD.module_code = 'FT' THEN $NOTIFY := 'Y'; ELSE $NOTIFY := 'N'; END IF; RETURN TRUE;
```

Web Service Tags

□	Order	Xsd Field	Table Field	Data Type	Maximum Length
□	1	SOURCEREFNO	EXTERNAL_REF_NO	VARCHAR2	
□	2	CONTREFNO	CONTRACT_REF_NO	VARCHAR2	
□	3	BOOKDT	BOOK_DATE	DATE	

d. Modify the details as required and click on the **Save** icon.

   Selected RADXML file gets modified.

8

Notification Flow

This topic describes the run time Notification flow.

Notification flow

The notification process occurs in two parts:

1. Oracle JOBs created using FCJ Scheduler framework that sends data required for notification to an internal JMS queue.
2. Gateway MBD that lists on internal JMS queue, that picks the notification XMLs and prepares full web service response and send to external system queues.

Scheduler Based Notification

The Notification Process in FLEXCUBE can be done using the jobs scheduler as follows:

1. The trigger generated from Workbench will be inserting key details into a static notification log (STTB_NOTIFICATION).
2. Once Job is triggered, a request is sent to the EJB layer from the job execution class and the notification log table will be polled for unprocessed records.
3. Each unprocessed record is locked. The record is verified against the notification maintenance and checked whether notification is to be sent or not.
4. If notification is to be sent, pre-notification message XML is built and it is sent to internal NOTIFY_QUEUE(JMS queue) configured in Gateway layer.
5. The job is then rescheduled to fire next time based on the previous execution.

Refer Gateway Installation documents on how to set up the Queues.

Figure 8-1 Architecture- Notification Flow in Scheduler

MDB Based Notification Flow

Notification processes in MDB are as follows:

1. Notification MDB listens on the internal NOTIFY_QUEUE(JMS queue).
2. On any message received, the MDB identifies which schema to connect using the JNDI name being present as part of the message XML.
3. Gateway notification processing package is called from MDB to build notifications.
4. In MDB, the notifications built are processed and sent to the destination specified in the corresponding notification.
5. In case of an exception the transaction is rolled back.
6. If all notifications are successfully processed, the transaction is committed.

9 Test Notification

This topic provides systematic instructions to test the notification.

1. Simulate a case where the base table undergoes data change.
2. Check record populated at **STTB_NOTIFICATION** table
3. Check Notification message: **GWTBS_NOTIFICATIONS_LOG.NOTIFICATION_MESSAGE**