Oracle FLEXCUBE Universal Banking ® 12.0.2 Development Workbench - Notifications

August 2013



Contents

1	Pref	ace3
1	.1	Audience3
2	Intro	oduction3
Н	low to	o use this Guide3
3	Noti	fication – Getting started4
3	.1	What is Notification
3	.2	Notification Trigger4
4	Noti	fication Development4
4	.1	Pre-request for Notification development and testing4
4	.2	Notification specification5
4	.3	Notification XML development5
4	.4	Notification Process
4	.5	Development process in Development Workbench
4	.6	Notification Trigger
4	.7	Notifications
5	Dep	loy Notification19
5	.1	Notification - Workbench related deployment19
5	.2	Notification Trigger deployment
6	Test	Notification
6	.1	Notification flow
6	.2	Scheduler based notification
6	.3	MDB based notification flow21
6	.4	Triggering notification and testing23

1 Preface

This document describes the steps to develop the notification XML and notification trigger using Oracle FLEXCUBE Development Workbench for Universal Banking.

1.1 Audience

The Development Workbench Notification Development book is intended for the FLEXCUBE Application Developers who perform the following tasks:

Develop new Notification

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

Proficiency	Resources
FLEXCUBE UBS Development	FCUBS-FD01-01-01-Development
overview	Overview Guide
Interface Getting started	FCUBS-FD04-01-01-Interface Getting started
FLEXCUBE Development Workbench for Universal Banking Reference	User manuals
Web service development to have query web service in place	FCUBS-FD02-03-01-RAD Web Service Development

2 Introduction

How to use this Guide

The information in this guide includes:

- <u>Chapter 3, "Introduction"</u>
- <u>Chapter 4, "Notification Getting started"</u>
- <u>Chapter 5, "Notification Development"</u>
- <u>Chapter 6, "Deploy Notification"</u>
- <u>Chapter 7, "Test Notification"</u>

3 Notification – Getting started

3.1 What is Notification

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

3.2 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

4.1 Pre-request for Notification development and testing

Following are pre-request 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



Fig 5.1.1: Development of Notifications

4.2 Notification specification

Identify the notification requirement as below

- What is the Notification function ID name for RAD XML (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 filter?
- What is the query Web service to be used?
 - What is the operation?
 - What are the tags required?

Example;

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

4.3 Notification XML development

Notification RAD XML development creates the following files:

- RAD XML
- SPC
- SQL
- Static Data

4.4 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.

4.5 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 to 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 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:**

4.6 Notification Trigger

After successful login to Development Workbench click on Notification Trigger option in the tree as shown below:





Notification Trigger				_ ×
		I I Z (<u>}</u> 7	(g) =\$
Trigger Code Description Firing Time	Base Table * PK Cols * PK Cols * PK Types * Van			
Selected Columns	Uts Uts Uts Notification Codes Image: Code State			
Trigger When Clause	votify to VN)			

Fig 4.6.2: Notification trigger options

Notification Trigger we have two options - Add a new Trigger or Modify Existing one.

New:

Notification Trigger						_ X
Notification Trigger Trigger Code * Description Firing Time Each Record	TRG_CUSTADDR Base Table * MSTM_CUST_ADDRESS Trigger for Customer Address Maintenand P PK Cols * CUSTOMER_NO-LOCATION-MEDIA After * PK Types * VARCHAR2-VARCHAR2-VARCHAR2 Yes * Data Types	×	I I	2	8	- ×
Selected Columns	Notification Codes					
Trigger Logic Sei SNC IF NVL(c LOpe ELSE LOPE END IF;	pTFFY To YN) oldonce_auth, 'N) ↔ 'Y THEN ration := 'UPDATE';					

Fig 4.6.3: Notification trigger: New option

Trigger Code: A unique value to for a notification trigger. Follow the naming conversion it should start with **TRG_XXXX**. This is mandatory field. This attribute signifies the trigger code created as part of trigger creation step in OTD. Each notification will be linked to a trigger code.

Description : Information field. Meaningful description of Trigger is to be given.

Firing Time : Specify when trigger needs to fired. We 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 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 mandatory field.

Pk Types: Enter Primary key type of the corresponding primary key field. This is 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 in order 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, field can be left empty

erate Rad Filos									
Front-End Files	System Packages	H	ook Paokagea		Meta Data			Others	^
] RadXML Screen Xml System JS	Main Package Spec Main Package Body Noblication Trggers Upload Package Spec Upload Package Body	Kernel P Kernel P Cluster F Cluster F Custom I Custom I	ackage Spec ackage Body 'ackage Spec 'ackage Body Package Spec Package Body	Menu Datalis Datasource Datalis LOV Details Block Datalis Screen Details Amendable Details Call form Defails Summary Details	Lt Bl G G V N Ft Pl	abel Detalis lock FK Columns unction Call Forms ateway Details oblication Details unction Parameters urge Details	Xads Xad With / Screen Hi Upload Ta Upload Ta Archive Ta	Annotations mi ible Trigger ibles Definition ble Definition	
SLNo		File Name			1	File Type	:	itatus	
GWT	R#_TRG_CUSTADDR Irg			F	TRG			*	
GWTI	M_NOTIFICATION_TRIGGERSTRG_C	USTADOR INC		Î	NC		Generalsd	*	
TRG	CUSTADDR RAD Imi			ĩ	RADXML			v	
F	ror Description	Fra	ar Code						
El Deguest suo	nor Description		SAVE 007						
			*					Generate	Exit

Fig 4.6.4: Notification trigger: Generation

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

Modify	:

Notification Trigger		_ ×
		📔 🗶 🖬 📝 🍃 🐬 🇐 🔿
Trigger Code *	Page Table *	
Description		
Description		
Firing Time	Vac w	
Each Record	Data Types	
Selected Columns	Notification Codes	
Trigger When Clause		
Trigger Logic(Set SN(
	×	
	Ψ.	
1		

Fig 4.6.5: Notification trigger: Modification

Notification Trigger					200	• ×
			×	Y YÍ		
	Trigger Code	×				
Trigger Code *	Trigger Code					
Description						
Fach Record Yes *						
Selected Columns	Search Reset					
Trigger When Clause]				
	<< < 1nf5 > >>					
Trigger Logic(Sel \$NOTIFY To Y/N)	Trigger Code					
	TD AC BRN TFR					
	TRG_APP_DETAIL					
	TRG_BLKDELMSTR					
	TRG_BRTMMSTR					
	TRG_CAIDET					
	TRG CHBK					
	TRG_CLAC					
	TRG_CLTMPRD					
	TRG_CONT					
	TRG CUST					
	· · · · · · · · · · · · · · · · · · ·					

Fig 4.6.6: Notification trigger: Modification-Selecting Trigger name

Notification Trigger					-
			×	I 17	1 F 🌒
				E	xecute Query
				_	
Trigger Code * TRG_CUSTADDR	A 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)					
		<u>^</u>			
		-			

Fig 4.6.7: Notification trigger: Modification- Entering values

Notification Trigger						-	×
	×	r	1	Ъ	Ŧ		⇔
Trigger Code * TRG_CUSTADDR Base Table * MSTM_CUST_ADDRESS							
Description Trigger for Customer Address Maintenand PK Cols * CUSTOMER_NO-LOCATION-MEDIA							
Firing Time After * PK Types * VARCHAR2-VARCHAR2-VARCHAR2							
Each Record Yes Data Types							
Selected Columns Notification Codes							
Trigger When Clause (new.auth_stat='A')							
Trigger Logic (Set SNOTEV To VN.)							
IF NVL(old.once_auth, N) ↔ Y THEN							
LOperation := INSERT;							
LOPeration := "UPDATE";							
END IF:							

Fig 4.6.8: Notification trigger: Modification- Entering values

Front-En	d Files System Packages	Hook Paokages	М	eta Data	Others	<u> </u>
] RadXML] Screen Xml] System JS	Main Package Spac Main Package Body Violotication Triggers Upload Package Spac Upload Package Body	Kernel Package Spac Kernel Package Body Cluster Package Spac Cluster Package Spac Custom Package Spac Custom Package Body Custom Package Body	Menu Detalis Datasource Detalis LOV Detalis Biock Detalis Screen Detalis Amendable Detalis Call form Detalis Summary Detalis	Label Datalis Block FK Columns Function Call Forms Caleway Details Notification Details Function Parameters Purge Details	X3d3 X3d With Annotations Screen Himi Upload Table Tropper Upload Tables Definition Archive Table Definition	
SLNo		File Name		File Type	Status	^
	GWTR#_TRG_CUSTADDR Irg		TRG	;	Generalad 👻	
	GWTM_NOTIFICATION_TRIGGERS_TRG	CUSTADDR INC	INC		Generalad	
	TRGCUSTADDR_RAD.sml		RAD	XML	Generaled +	
! Re	Error Description equest successfully Processed	Error Code ^ RD-SAVE-007				

Fig 4.6.9: Notification trigger: Modification-Successful Generation

4.7 Notifications

Notifications Screen will be used to create new notification or modify existing notification; here we capture notification information for notification codes. We save notification details into xml.

Notification Maintenance							- 7 g	× ⇔
No	tification Function		Action None -	Save XM	L Path			
Notification Code * Description Notification Xsd Firing Time Filter Type Gateway Service Gateway Operation Gateway IO Request Type XSD Name Filter Logic(Set \$NOTIFY To Y	▼) ▼ N & Refer Current Record as	\$CURRENT_RECORD)	Module Des Bas F Pł	Module				
Web Service Tags					+ -			
Order	Xsd Field	Table Field	Data Type	Maximum Length	~			

Fig 4.7.1: Notification Screen

Action: We can choose either new or Load action. New to create a new notification and Load 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 is specified as \$CURRENT_DIRECTORY

Notification Function: Specify the notification function-id name.

Conventions:

Maximum 8 chars. 3^{*rd*} *letter must be 'N'. Example: FTNCONON*

Notification Code: Enter the notification code to which we need to capture values. This is Mandatory field.

Recommended Convention for Notification Codes: NOTIF_<Module Code>_<Description> Example: NOTIF_LD_CONTRACT This is the notification indicating that a LD contract has been created/modified **Description:** Information field. 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 which would be defaulted from Module LOV

Notification XSD: Notification XSD name will have to be provided in the corresponding Field. Naming convention to be followed while naming Notification XSD is as follows

[Module Name] – [Notification Description] – Notif.xsd 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 has to be sent. Options available are Insert, Update or Both

Filter Type: This attribute can take the following values.

- 1. Where clause
- 2. 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 query for the mentioned Service.

Gateway Service: The gateway service to be used to get the full screen response.

Gateway IO Request: The gateway IO request node to be used in querying operation.

Type XSD Name: This field has to be entered if Notification is mapped to a Service and Request. Name of the Master Type XSD for the service and operation has to be provided here. This can be found in include portion of the Request Msg XSD of particular Service-Operation

Example: LC-Contract-Types.xsd

Full screen Reply: This attribute decides whether full screen or primary key notification response to be sent. This is applicable only if gateway Service details are provided

HO only: This attribute is used to send notification only from head office.

Filter Logic: The filter logic which decides whether the notification needs to be sent or not. This can be a simple where-clause on base table or a complex pl/sql block.

Web service Tags: The columns selected from base table as part of web service tags, will be used to send the full screen notification response. These tags defines the elements of Notification Xml when no Query service is mapped to it .

Front Find Films	Sustan Baskana	Hash Bashanas		Mata Data	Others	
RadXML Screen Xml System JS	System Packages V Main Package Spec V Main Package Body Notification Triggers Upload Package Spec Upload Package Body	Kernel Package Spec Kernel Package Body Cluster Package Body Cluster Package Body Custom Package Body Custom Package Spec Custom Package Body	Menu Details Datasource Details LOV Details Biock Details Screen Details Call form Details Summary Details	Meta Data Label Details Block PK Columns Function Call Forms Gateway Details Notification Details Function Parameters Purge Details	V Xsds Xsd With Annotations Screen Html Upload Table Trigger Upload Tables Definition Archive Table Definition	
SI.No	F	ile Name	File Type		Status	
					Generate	Exit



Notification M	aintenance	>					×
						x	19 🧃 🐳
Generate Rad	l Files						×
							A
Fr RadXML Screen) System	ront End File JS	s System Pa Main Package Sp Main Package Br Notification Trag Upload Package Upload Package	ckeges Heck Packages ecc Kemel Package Spec (dy Kemel Package Spec cluster Package Spec Cluster Package Spec Spec Cluster Package Body Body Cluster Package Body Cluster Package Body	Menu Delalis Datasource Delalis LOV Details Bilock Delalis Screen Details Amendable Details Call form Delalis Surmary Details	Mets Data Label Defails Block PK Columns Function Call Forms Galeway Defails Notification Defails Function Parameters Purge Defails	Volteres Xsds Screen Himi Upload Table Trigger Upload Table Crefinition Archive Table Crefinition	
SIN	ia		Cile Namo		Eila Tuno	Status	*
1		fipks_finconon_main.spc			SPC	Generated *	
2		fipks_finconon_kernel.spc			SPC	Generated 👻	
3		fipks_ftnconon_main.sql	Information		×	Generated 👻	
4		fiplds_Rnconon_kernel.sql				Generated 👻	
5		GWTM_NOTIFICATIONS_MAST	Error Description		Error Code	Generated 💌	
6		GWTM_NOTIFICATION_TAG_N	 Request successfully Processed 		RD-SAVE-007	Generated 👻	
7		FTNCONON_RAD xml				Generated	
					-		-
					Ok	Generate	Exit
		Do you want to open or save	RAD.ZIP from 10.184.132.100?		Open Sav	e 🔹 Cancel 🗙	

Fig 4.7.3: Notification Screen Generation Successful

Modifying an Existing Notification RADXML

The process of modifying an existing Notification RADXML is illustrated in the images below

Notifica	tion Main	tenance																					(1 43	 ×
			Nolificatio	on Functio	m					Action	n Load	-				Lo	ad Scree	en Xmil			BI	ROWSE]	
	Noblicat	ion Code *											Mo	dule [
Uplo	es. ad	noristion					×		<i>(</i>	Choose Fi	ile to Uplo	pad		P							x	1		
) • M	AIN 🕨 F	T 🕨 RAE	XML		•	47	Search RA	DXML		٩			
		Lipload E	ilo			Brows)rganize	▼ Ne	ew folder								•	0			
		Opioau P	lie			DIOWS				Dov	vnloads	*	Name		^			Date r	nodified	1	Гуре 📩			
										Tec Rec	ent Places			FTDUP_R	AD.xml			12/23	/2012 9:4	8 AM				
								RECOR		Deskt	op		FTC	PRJET_RA	D.xml			12/23/	/2012 9:4	8 AM	KML F			
								5		Libr	aries raju	=	FTD	CAUTH_F	RAD.xml			12/23	/2012 9:4	8 AM	KML F			
										r Con	nputer		FTD	CONAU_I	RAD.xml			12/23,	/2012 9:4 /2012 9:4	8 AM 3 8 AM 3	KML F KML F			
										≚ Sy Da	/stem (C:) ata (D:)		FTD	CXFRA_R	AD.xml			12/23	/2012 9:4	8 AM 3	KML F			
										B D	VD RW Dri	ivel	FTDI	NCKCH ^T	RAD.xml			12/23,	/2012 9:4 /2012 9:4	8 AM 3 8 AM 3	KML F KML F			
										FL	.EXCUBE_I peria L	Ken	FTD	MT101_R	AD.xml			12/23	/2012 9:4	8 AM 3	KML F 🔻			
												File nan	ne:				▼ A	ll Files (*.*)		-			
																		Open		Cance				
																		-				J		
				Xsd Fi	ield																			

Fig 4.7.4: Notification Screen Loading

										.	×	V	
			Notification Function	TNCONON	Actio	n Load 🔻	Save X	ml Path FTNCONO	N_RAD.xml	BROWSE			
1	Notific	ication Code	* NOTIF_FT_CONTRACT			Module	FT]				
	I	Description	This is the notification in	ndicating that a FT 🖻		Module Description	Funds Transfer						
	Notif	tification Xsd				Base Table	* CSTB_CONTRACT	=					
	ſ	Firing Time	Insert 👻			PK Cols	* CONTRACT_REF_NO						
		Filter Type	Pisql Block 🛛 👻	_		PK Types	* VARCHAR2						
	Gatev	eway Service	FCUBSFTService	<u></u>			Full Screen Reply						
Ga	Gateway	ay Operation	QueryContract	×=			HO Only						
Gat	ateway I	/ IO Request	Contract-Details-IO										
	Type 2	e XSD Name											
Filter L	Logic(: IF	(Set SNOTIF IF SCURREI	Y To YIN & Refer Current R	Record as \$CURRENT_RECORD) E \$NOTIFY := 'N';	END IF; RETURN TRUE;	•						
ilter L	Logic(:	(Set \$NOTIF	Y To Y/N & Refer Current R	Record as \$CURRENT_RECORD) E \$NOTIFY := 'N';	END IF; RETURN TRUE;							
ilter L	Logic(: IF	(Set \$NOTIF IF \$CURREI Service Ta	Y To Y/N & Refer Current R IT_RECORD.module_cod	Record as \$CURRENT_RECORD) E \$NOTIFY := 1V;	END IF; RETURN TRUE;		+-					
V	Web S	(Set \$NOTIF IF \$CURREI Service Ta rder	Y To Y/N & Refer Current R IT_RECORD.module_cod	Record as \$CURRENT_RECORD le = TT THEN \$NOTIFY := Y; ELS Table Field) E \$NOTIFY := 1V;	END IF; RETURN TRUE; Data Type	Maximum Leng	+ -					
V 1 1	Web S	(Set \$NOTIF IF \$CURREI Service Ta rder	Y To Y/N & Refer Current R IT_RECORD.module_cod Igs Xsd Field SOURCEREFNO	Record as SCURRENT_RECORD le = TT THEN \$NOTIFY := Y; ELS Table Field EXTERNAL_REF_NO) E \$NOTIFY := 1N; I	END IF; RETURN TRUE; Data Type VARCHAR2	Maximum Leng	+ -					
V	Web S Orc 1	Service Ta	Y To Y/N & Refer Current R IT_RECORD.module_cod IT_RECORD.module_cod SOURCEREFNO CONTREFNO	Record as \$CURRENT_RECORD ie = TT THEN \$NOTIFY := Y; ELS Table Field EXTERNAL_REF_NO CONTRACT_REF_NO) E \$NOTIFY := 1V; I PE PE	END IF; RETURN TRUE; Data Type VARCHAR2 VARCHAR2	Maximum Leng	th A					

Fig 4.7.5: Notification Screen Loaded

Niat												
Not										×	V	9
Not		Notification Function	TNCONON	Actio	n Load 👻	Save Xml	Path FTNC	NON_RAD.xml	BROWSE			
110	tification Code *	NOTIF_FT_CONTRACT	-		Modul	e FT		* E				
	Description	This is the notification in	ndicating that a FT 🖻		Module Description	Funds Transfer						
N	lotification Xsd				Base Table	e * CSTB_CONTRACT						
	Firing Time	Insert 👻			PK Col	* CONTRACT_REF_NO						
	Filter Type	Plsql Block 🔹	_		PK Type:	s * VARCHAR2						
Ga	ateway Service	FCUBSFTService	* E			Full Screen Reply						
Gater	way Operation	QueryContract	×=			HO Only						
Gatew	ay IO Request	Contract-Details-IO										
Ту	pe XSD Name											
Mol	h Coprise Top	-				÷	•					
Wel	b Service Tag	s	7.11.5			Ŧ	+ -					
Wel	b Service Tag Order	s Xsd Field	Table Fie	Id	Data Type	Maximum Length	+-					
Wel	b Service Tag Order	s Xsd Field OURCEREFNO	Table Fie	ld ri	Data Type VARCHAR2	Maximum Length						
Wei	b Service Tag Order	s Xsd Field DURCEREFNO ONTREFNO	Table Fie EXTERNAL_REF_NO CONTRACT_REF_NO	ld rij	Data Type VARCHAR2 VARCHAR2	 Maximum Length 64 7 						

Fig 4.7.6: Notification Screen Loaded and Modified

5 Deploy Notification

5.1 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

5.2 Notification Trigger deployment

Compile the following files in Target FLEXCUBE UBS Database schema

- GWTM_NOTIFICATION_TRIGGERS__TRIG_CONTRACT.INC
- GWTR#_TRIG_CONTRACT.TRG

6 Test Notification

This section explains the run time notification flow and testing steps.

6.1 Notification flow

The notification process occurs as 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 prepare full web service response and send to external system queues.

6.2 Scheduler based notification

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

The trigger generated from Workbench will be inserting key details into a static notification log (STTB_NOTIFICATION)

Once Job is triggered, a request is sent to EJB layer from job execution class and the notification log table will be polled for unprocessed records.

Each unprocessed record is locked.

The record is verified against the notification maintenance and checked whether notification is to be sent or not.

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.

The job is then rescheduled to fire next time based on the previous execution.

Refer Gateway Installation documents on how to setup the Queues.

Flow Chart for Notification Flow in Scheduler



6.3 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 is processed and sent to the destination specified in corresponding notification.
- 5. In case of exception the transaction is rolled back.
- 6. If all notifications are successfully processed, transaction is committed.

Flow Chart for Notification Flow in MDB



6.4 Triggering notification and testing

Follow the below steps to test notification

- Simulate a case where base table under goes data change.
- Check record populated at STTB_NOTIFICATION table
- Check Notification message
 GWTBS_NOTIFICATIONS_LOG.NOTIFICATION_MESSAGE



Development Workbench - Notifications August 2013

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/

Copyright © 2012-2013 Oracle Financial Services Software Limited. All rights reserved.

No part of this work may be reproduced, stored in a retrieval system, adopted or transmitted in any form or by any means, electronic, mechanical, photographic, graphic, optic recording or otherwise, translated in any language or computer language, without the prior written permission of Oracle Financial Services Software Limited.

Due care has been taken to make this *Development Workbench-Notifications* and accompanying software package as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this *Development Workbench-Notifications* and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.

All company and product names are trademarks of the respective companies with which they are associated.