Development Workbench - Notifications Oracle FLEXCUBE Investor Servicing Release 14.7.2.0.0 Part No. F75930-01 [May] [2023]





Table of Contents

1.	Prefa	ace3
1	.1	Audience
2.	Intro	duction4
2	.1	How to use this Guide4
3.	Notif	ication – Getting started5
3	.1	Notification Development5
	3.1.1	Pre-request for Notification development and testing5
3	.2	Notification specification
3	.3	Notification XML development
3	.4	Notification Process
3	.5	Development process in Development Workbench
3	.6	Notification Trigger7
3	.7	Notifications
4.	Depl	oy Notification
4	.1	Notification - Workbench related deployment17
4	.2	Notification Trigger deployment
5.	Test	Notification
5	.1	Notification flow
5	.2	Scheduler based notification
5	.3	MDB based notification flow
5	.4	Triggering notification and testing21

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 overview	FCIS-FD01-01-01-Development Overview Guide
Interface Getting started	FCIS-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	FCIS-FD02-03-01-RAD Web Service Development

2. Introduction

2.1 How to use this Guide

The information in this guide includes:

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

3. Notification – Getting started

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.

Notification Trigger

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

3.1 Notification Development

3.1.1 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: Development of Notifications

3.2 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 UTNUH
- Notification code NOTIF_UNITHOLDER
- Base table UNITHOLDERTBL
 - Operation DELETE
 - Filter Account class type in (S, U)
- Web service to be used FCISUnitholder
 - Operation QueryUnitholder
 - Request node Unitholder-IO

3.3 Notification XML development

Notification RAD XML development creates the following files:

- RAD XML
- SPC
- SQL
- Static Data

3.4 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.

3.5 <u>Development process in Development Workbench</u>

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

3.6 Notification Trigger

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

🍯 ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox	-		×
O 🖄 🕫 Or whf00igw:8088/FCJRADTool142GA/RADLoginServlet#nogo		☆	≡
ORACLE FLEXCUBE Development Workbench for Universal Banking		Adr	nin
X Windows	Options	Sign Ou	ut
Administration Function Generation Screen Customizer Tracking Changes Notifications Buik Generation Reset Buik Generation Refresh Block Detail Upload Test Case Definition Nuchart Definition Test Case Data Upload Create Request Nuchart Definition Runchart Execution Request Nuchart Definition Request Request Nuchart Definition Request Request Request Nuchart Definition Request			

Fig: Notification trigger

CRACLE FLEXCUBE Developm	ient Workbench for Universal Banking — Mozilla Firefox		-		×
🔿 👌 🗝 👓 whf00iqw	8088/FCJRADTool142GA/RADLoginServlet#nogo			☆	=
ORACLE' FLEXCUBE	Development Workbench for Universal Banking			Adm	nin
X Notification Trigger		Windows	Options	Sign Ou	t ×
in an		2 🗵 ! 🔟	<u> </u>	1 7 🤤	- ^
		_			
Trigger Code • Description Fining Time Each Record Selected Columns Trigger When Clause	Base Table • PK Cols • PK Types • Data Types • Data Types • Notification Codes				
Trigger Logic (Set \$NOTIF)	To YM) (Trigger Logic)				

Fig: Notification trigger options

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

New:

ORACLE FLEXCUBE Development Workbench for Universal Ba	nking — Mozilla Firefox				-		×
🔿 🖄 🕶 🕶 whf00iqw:8088/FCJRADTool142GA/RA	DLoginServlet#nogo					\$? ≡
DRACLE' FLEXCUBE Development Workbench fe	or Universal Banking					Ac	dmin
×				Window	s Options	Sign	Out
lotification Trigger				a 🗙 I I	2	1	- ()
Trigger Code • TRG UHCREATION		Base Table * UNITHOLDERTBL	P				
Description Trigger On Unitholder Table for ne	w and modify 🖻	PK Cols * UNITHOLDERID					
Firing Time After Ves V		PK Types * VARCHAR2					
Selected Columns		Data Types					
Trigger When Clause (new.auth_stat='A')	P	Nonicatori Codes Norin _DiarrioEDER					
Trigger Logic (Set \$NOTIFY To Y/N)							
IF <u>NVL</u> (:old.once_auth, 'N') <>'Y' THEN Loperatrion:='INSERT';							
ELSE I_operatrion:='UPDATE';							
END IF;							
		ĥ.					



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

FLEAUDE	Development Workbench for U	niversal Banking				A
					Windows Options	Sign
ition Trigger						
					🗎 🗶 🗉 🖽 🦕	13
	Upload Package Spec	Cluster Package Body	Screen Details	Gateway Details Notification Details	Archive Table Definition	
	Java Classes	Custom Package Body	Amendable Details	Function Parameters		
	Information			Purge Details		
	Information Description		Information	ELCM MetaData Class		
			Code			
	Request successfully Pr	ocessed				
SI.No				File Type	Status	^
GW	R#			TRG	Generated v	
GWI				101m	Constant and	
GWT	M			INC	Generated	
GWTI GWTI TRG_	M			RADXML	Generated V	
GWTI GWTI TRG_	MUF			RADXML	Generated V	
GWTI GWTI TRG_	M		Ok	RADXML	Generated ×	
GWTI GWTI TRG_	M	_	Ok	RADXML	Generated	
GWT	M		Ok	INC RADXML	Generated v	
GWTI GWTI TRG_			OK	RADXML	Generated ×	
GWII GWII TRG_			Ok	RADXML	Generated V	
GWII GWII TRG			Ok	INC RADXML	Generated V	
GWTI TRG			Ok.	INC RADXML	Generated ×	
GWTI TRG			Ok	INC RADVML	Generated V Generated V	

Fig: 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:

ACLE FLEXCUBE Develop	iment Workbench for Universal Banking — Moz	illa Firefox				-	
& ≅ ∽ whf00iq	w:8088/FCJRADTool142GA/RADLoginServ	/let#nogo					☆
CLE FLEXCUBE	Development Workbench for Universal	Banking					Adn
					Windows	Options	Sign Ou
ation Trigger							
					2 ×	13 1	2
Trigger Code	•	1	Base Table •	P			
Description			PK Cols *				
Firing Time	Before v		PK Types *				
Each Record	Yes v	1	Data Types				
Selected Columns			Notification Codes				
Trigger When Clause							
Trigger Logic (Set \$NOTI	FY To Y/N)						
			li.				



LEXCODE Developi	nent Workbench for Universal Banking		Admi
4)	-		Windows Options Sign Out
uon ingger			
Trigger Code *	P	Base Table *	
Description	P	PK Cols *	
Each Record Yes 👻		Data Types	
Trigger When Clause	Code	trication Codes	
	Trigger Code		
igger Logic(Set \$NOTIFY 1	Se	arch Reset	
	Trigger Code		
TRG	UHCREATION		
		1.	
		di.	

Fig: Notification trigger: Modification- Selecting Trigger name

🥮 ORACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox		-		\times
O 🖄 🕫 ⁰ฯ wh100iqw:8088/FCJRADTool142GA/RADLoginServlet≢nogo			ŝ	≡
ORACLE: FLEXCUBE Development Workbench for Universal Banking			Adr	nin
X	Windows	Options	Sign O	ut
Notification Trigger				- ×
	X	12 🦢	121	
		Execute	Query	
Trigger Code + TRG_UHCREATION Base Table +				
Description PK Cols *				
Firing Time Bellet PK Types Each Record Yes Data Types				
Selected Columns Notification Codes				
Trigger When Clause				
Trigger Logic (Set \$NOTIFY To Y/N)				

Fig: Notification trigger: Modification- Entering values

SRACLE FLEXCUBE Development Workbench for Universal Banking — Mozilla Firefox				-		×
O 👌 ≅² ⊶ whf00iqw:8088/FCJRADTool142GA/RADLoginServlet#nogo					ŝ	=
ORACLE' FLEXCUBE Development Workbench for Universal Banking					Ac	lmin
X Notification Trigger	_	Windo	ws	Options	Sign	Dut
			I	of 🚡	1	- ^ 9 🔿
Trigger Code TRG_UHCREATION Base Table * UNTHOLDERTBL Description Figger Col Untholder Table for new and modify PK Firing Time After v PK Types * UNTHOLDERID Each Record Yes v Data Types Selected Columns Notification Codes NOTF_UNTHOLDER Trigger Logic (Set \$NOTFY To YN) F MULCidd conc. sub; NY >>* TriEN Logenzation = NESET;						
ELSE Logeration = UPDATE; END F;						

Fig: Notification trigger: Modification- Entering values

CLE FLEXCUBE	Development Workbench for Universal Banking				Ac
				Windows Ontions	Sign (
ation Trigger				Windows Options	Sign
					18
	Upload Package Spec. Cluster Package Body Upload Package Body Custom Package Spec. Java Classes Custom Package Body	Block Details Creen Details Amendable Details	Gateway Details Notification Details	Upload Tables Definition Archive Table Definition	
	Information		Purge Details		
	Information Description	Information Code	ELCM MetaData Class		
	Request successfully Processed				~
SI.No			File Type	Status	~
GWT	2.#		TRG	Generated V	
GWT	<u>a</u>		INC	Generated V	
TRG	0+		RADXML	Generated V	
		Ok			



3.7 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.

ORACLE FLEXCUBE Develop	oment Workbench for Universal Ban	king — Mozilla Firefox				-		×
🔿 👌 🗝 👓 whf00iq	w:8088/FCJRADTool142GA/RAI	DLoginServlet#nogo					☆	=
ORACLE FLEXCUBE	Development Workbench for	r Universal Banking					Ad	min
=					Windows	Options	Sign (Dut
Notification Maintenance								_ >
							17	₫ 🗢
	Notification Function		Action New 🗸	Save XML Path				
Notification Code	•		Module		P			
Description		2	Module Description					
Notification Xsd			Base Table	*				
Firing Time	~		PK Cols	•				
Filter Type	¥		PK Types	•				
Gateway Service				Full Screen Reply				
Gateway Operation				Specific Notification				
Type XSD Name								
Filter Logic (Set \$NOTIEX To	VIN & Refer Current Record as \$CL							
The Logic (oct ofform 1 to		Intern_neoonb /						
				10.				
Web Service Tags				+ -				
Order	Xsd Field	Table Field	Data Type	Maximum Length				
			2					

Fig: 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. 3rd 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.

	boroopinent frombenen for e	ninversai isanning				Aun
					Windows Options	Sign Ou
erate Rad Files						
Front-End Files	System Packages	Hook Packages		Meta Data	Others	^
RadXML Screen Xml System JS	Man Package Spéc Man Package Body ✔ Notfication Triggers Upload Package Spec Upload Package Body Java Classes	 Kernel Package Spec Kernel Package Body Cluster Package Spec Cluster Package Body Custom Package Spec Custom Package Body 	Menu Detais Datasource Detais LOV Detais Biock Detais Screen Detais Armendable Detais Call form Detais XSD Detais XSD Details	Label Details Block PK Columns Function Call Forms Gateway Details Motification Details Function Parameters Purge Details ELCM MetaData Class	Xada Screen Html Upload Tables Definition Upload Tables Definition Archive Table Definition	~
SI.No		File Name	Fi	ile Type	Status	^



Front-End F	Files	System Packages	Hook Packages		Meta Data	Others		
RadXML Screen Xml System JS		Man Package Bopc Kernel Package Body Kernel Package Body Kottication Triggers Upload Package Spec Upload Package Spec Upload Package Spec Jupad Package Body Custer Package Spec Java Classes Custom Package Body Information		Menu Details Datasource Details LOV Details Block Details Screen Details Amendable Details	Lade Detais Bock PK Columns Function Call Forms Gateway Details Votification Details Function Parameters Function Parameters ELCKM MetaData Class	AsdS Screen Html Upload Table Trigger Upload Tables Definition Archive Table Definition		
		Information Description C Request successfully Processed		Information Code				
SI.No	[dillo				File Type	Status		
	utpks_u				epc	Generated x		
	utoks_u				SOL	Generated Y		
	utoks u				SOL	Generated ~		
	undefine			Ok	XSD	Generated ~		
	GWTM	NOTIFICATIONS MASTER UTN	JHINC		INC	Generated V		
	UTNUH	RAD.xml			RADXML	Generated V		

Fig: Notification Screen Generation Successful

Modifying an Existing Notification RADXML

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

ORACLE FLEXCUBE Development Workbench for Universal Banking O & デ 0・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	Mozilla Firefox Servlet#nogo				_	ි Agr	× ≡
=				Windows	Options	Sign Ou	ut
Notification Function	Action	~	Load Screen Xml	BROWS			
Notification Code • Description Notification Xsd Fring Time Filter Type Gateway Operation Gateway IO Request Type XSD Name Filter Logic (Set \$NOTIFY To Y/N & Refer Current Record as \$CURREN	Se No file selected.	Module Module Description Base Table * PK Cols * PK Tunes * > FRONTEND > UIXML > ENG Name SCDDCULxml SCDDCULxml SCDDCULxml SCDDCUVxml SCCDDCVYxml SCCDDCVYxml SCCDDCVYxml	✓ ♥ ♥ P Search Date mod 11/16/202 11/16/202 11/16/202 11/16/202 11/16/202 11/16/202 ✓ All Files (*.*)	× ENG Iffied Type ↑ 2 533 PM XMLI 2 533 PM XMLI			
Web Service Tags Order Xsd Field	Table Field Data Typ	e Maximum I	Qpen	Cancel			

Fig: Notification Screen Loading

🔘 ORAG	CLE FLEXCUBE Devel	opment Workbench for Univ	ersal Banking — Mozilla Firefox						-		×
08	ল≏ ⊶ whf00i	igw:8088/FCJRADTool142	GA/RADLoginServlet#nogo							5	≡
URM			encinity onliversal ballking							Aar	nın
≡								Windows	Options	Sign O	ut
									X	V 🧃) 🔿
		Notification Function UTN	JH	Action	Load V	Save >	ML Path UTNUH_RAD.xml	BROWSE			
	Notification Code	*NOTIF_UNITHOLDER			Mode	e UT	Q				
	Description	This is the notification inco	licating that a Unithod 🖻		Module Descripti	n Unit Trust					
	Notification Xso	UNITHOLDERNotif.xsd			Base Tal	e * UNITHODLERTBL	P				
	Firing Time	e Insert v			PK C	s * UNITHOLDERID					
	Filter Type	Pisql Block v			РК Тур	s * VARCHAR2					
	Gateway Service	e FCISUnitholder	Q			Full Screen Reply					
	Gateway Operation	QueryUnitholder	Q			HO Only					
	Gateway IO Reques	t Unitholder-IO									
	Type XSD Name	5									
	IF \$CURRENT	RECORD.module_code='U	" THEN \$NOTIFY:='Y' ELSE \$NOT	TIFY:='N'; END IF;	RETURN TRUE;						
						li.					
	Web Service Tags						+ -				
	Order	Xsd Field	Table Field		Data Type	Maximum Length	^				
	1 SO	URCEREFNO	EXTERNAL_REF_NO	Q	VARCHAR2	64					
	2 CO	NTREFNO	CONTRACT_REF_NO	Q	VARCHAR2	7					
	3 BO	OKDT	BOOK_DATE	Q	DATE	64					

Fig: Notification Screen Loaded and Modified

4. Deploy Notification

4.1 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

4.2 Notification Trigger deployment

Compile the following files in Target FLEXCUBE UBS Database schema

- GWTM_NOTIFICATION_TRIGGERS__TRIG_CONTRACT.INC
- GWTR#_TRIG_CONTRACT.TRG

5. Test Notification

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

5.1 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.

5.2 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



5.3 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

5.4 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 [May] [2023] Version 14.7.2.0.0

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/

Copyright © [2007], [2023], Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.