

Notification Development
Oracle FLEXCUBE Investor Servicing
Release 14.1.0.0.0
[May] [2019]



Table of Contents

1. PREFACE	1-1
1.1 AUDIENCE	1-1
1.2 RELATED DOCUMENTS	1-1
1.3 CONVENTIONS	1-1
2. INTRODUCTION	2-1
2.1 HOW TO USE THIS GUIDE	2-1
3. NOTIFICATION - GETTING STARTED	3-1
3.1 WHAT IS NOTIFICATION	3-1
3.2 NOTIFICATION TRIGGER	3-1
4. PRE-REQUEST FOR NOTIFICATION DEVELOPMENT AND TESTING	4-1
5. NOTIFICATION DEVELOPMENT	5-1
5.1 NOTIFICATION SPECIFICATION	5-1
5.2 NOTIFICATION XML DEVELOPMENT.....	5-2
5.2.1 <i>Development Steps</i>	5-2
5.2.2 <i>OPEN DEVELOPMENT XML</i>	5-3
5.2.3 <i>SPC</i>	5-3
5.2.4 <i>SQL</i>	5-4
5.2.5 <i>Static Data</i>	5-4
5.2.6 <i>Verification Log</i>	5-4
5.3 NOTIFICATION TRIGGER DEVELOPMENT	5-4
5.3.1 <i>Development steps</i>	5-4
5.3.2 <i>Files</i>	5-7
5.4 DEPLOY NOTIFICATION	5-7
5.4.1 <i>Notification Open Development Related Deployment</i>	5-7
5.5 NOTIFICATION TRIGGER DEPLOYMENT	5-8
5.6 TEST NOTIFICATION	5-8
5.6.1 <i>Notification Flow</i>	5-8
5.6.2 <i>Scheduler Based Notification</i>	5-8
5.6.3 <i>MDB Based Notification Flow</i>	5-9
5.6.4 <i>Triggering Notification and Testing</i>	5-9

1. Preface

This document describes the steps to develop the notification XML and notification trigger using Open Development.

1.1 Audience

The Open Development 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 IS Development overview	Overview Guide
Interface Getting started	Interface Getting started
Open Development Getting started	Getting Started
Open Development reference	Reference
Web service development to have query web service in place	Web Service Development

1.2 Related Documents

For more information on FLEXCUBE IS environment, refer the following documents

- Overview Guide
- Interface Getting started
- Getting Started
- Reference
- Web Service Development

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or particular values. placeholder variables for which you supply.

monospace	Monospace type indicates language and syntax elements, directory and file names, URLs, text that appears on the screen, or text that you enter.
-----------	---

2.1 **How to Use this Guide**

The information in this guide includes:

- [Chapter 2, “Introduction”](#)
- [Chapter 3, “Notification - Getting started”](#)
- [Chapter 4, “Pre-request for Notification development and Testing”](#)
- [Chapter 5, “Notification Development”](#)

3. Notification - Getting Started

3.1 What is Notification

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

3.2 Notification Trigger

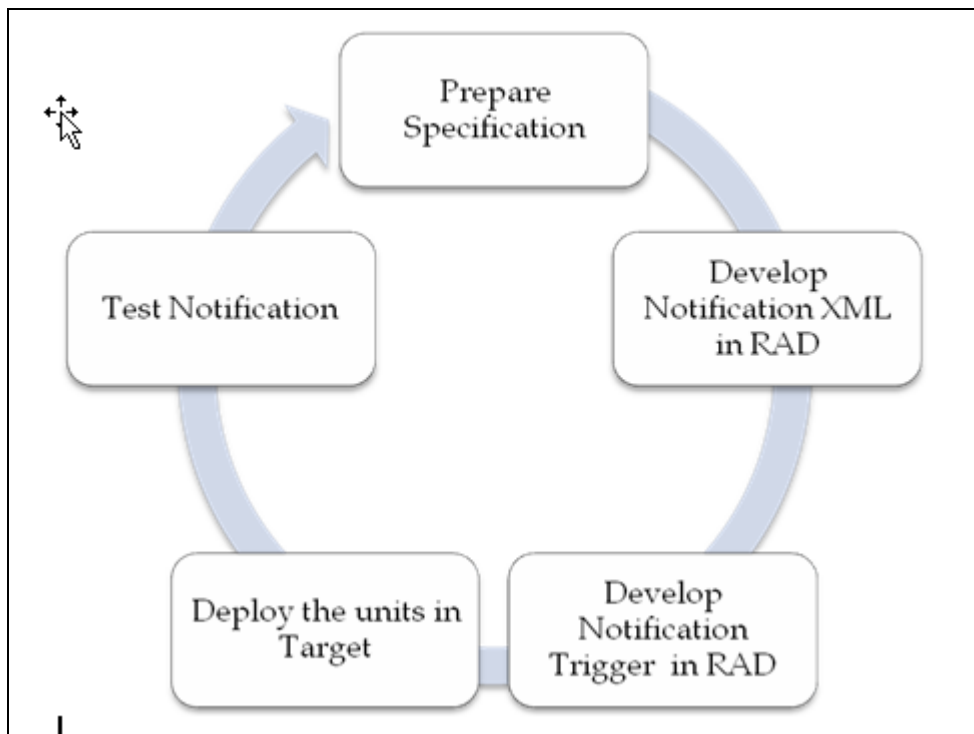
Notification Triggers is developed to recognize the event and then invoke the notification process. This trigger is developed using extensible Open Development Tool.

4. Pre-Request for Notification Development and Testing

Following are pre-request for notification development:

- Target FLEXCUBE Environment with Notification framework installed
- Open Development environment
- Required Query Web services developed and tested

5. Notification Development



5.1 Notification Specification

Identify the notification requirement as below

- What is the Notification function ID name for Open Development XML (Third character should be N)?
- What is the Notification code?
- What is the Base table in FLEXCUBE IS 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
 - Filter – Account class type in (S , U)
- Web service to be used - FCUBSAccService
 - Operation - QueryCustAcc

- Request node - Cust-Account-IO

5.2 **Notification XML Development**

Notification Open Development XML development, create the following files:

- Open Development XML
- SPC
- SQL
- Static Data

5.2.1 **Development Steps**

- Login extensible Open Development tool and select “Notification” from browser menu
- Select the Action “New”

The screenshot displays the Oracle Flexcube Management and Integration Console in a web browser. The user is logged in as Biswajit Mondal. The main menu includes 'Browser', 'Windows', 'Launch FC UBS', 'User Preferences', 'Change Password', and 'Sign Out'. The 'Notification Maintenance' form is active, showing fields for 'Action' (set to 'New'), 'Save XML Path', and 'Notification Function'. The form includes sections for 'Notification Code', 'Description', 'Operation' (set to 'Insert'), 'Filter Type' (set to 'None'), 'Gateway Operation', 'Gateway Service', and 'Gateway IO Request'. There are also fields for 'Module', 'Module Description', 'Base Table', 'PK Cola', and 'PK Types'. A 'Filter Logic' section contains the text: 'Set NOTIFY To Y/N & Refer Current Record as CURRENT_RECORD'. At the bottom, there is a 'Web Service Tags' table with columns for 'Order', 'Xsd Filed', 'Table Field', and 'Data Type'.

- Provide details as per the specification prepared earlier.



Refer Reference for meaning of each item in screen

Order	Xsd Filed	Table Field	Data Type
1	CUSTOMER_NO	CUSTOMER_NO	VARCHAR2
2	LOCATION	LOCATION	VARCHAR2
3	MEDIA	MEDIA	VARCHAR2

- Save and Generate. Further section explains the files that would be created.

5.2.2 Open Development XML

Open Development XML would be created as Notification Function ID>_Open Development.xml

Example

LDNTRG_Open Development.xml

5.2.3 SPC

Following SPC files would be created

- <module>pkcs_<Notification Function ID>_main.spc
- <module pkcs_<Notification Function ID>_kernel.spc
- <module pkcs_<Notification Function ID>_custom.spc
- <module pkcs_<Notification Function ID>_cluster.spc

Example

- Idpkcs_Idntrg_main.spc
- Idpkcs_Idntrg_kernel.spc

- ldpks_ldntrg_custom.spc
- ldpks_ldntrg_cluster.spc

5.2.4 **SQL**

Following SQL files would be created

- <module pks__<Notification Function ID>__main.sql
- <module pks__<Notification Function ID>__kernel.sql
- <module pks__<Notification Function ID>__custom.sql
- <module pks__<Notification Function ID>__cluster.sql

Example

- ldpks_ldntrg_main.sql
- ldpks_ldntrg_kernel.sql
- ldpks_ldntrg_custom.sql
- ldpks_ldntrg_cluster.sql

5.2.5 **Static Data**

Following INC files would be created

- GWTM_NOTIFICATION_TAG_MAP___<Notification Function ID>_.INC
- GWTM_NOTIFICATIONS_MASTER___<Notification Function ID>_.INC
- VARIABLE_MAPPING___<Notification Function ID>_.INC

Example

- GWTM_NOTIFICATION_TAG_MAP__LDNTRG.INC
- GWTM_NOTIFICATIONS_MASTER__LDNTRG.INC
- VARIABLE_MAPPING__LDNTRG.INC

5.2.6 **Verification Log**

Open Development creates the log file <Notification Function ID>_LOG.txt for generation status.

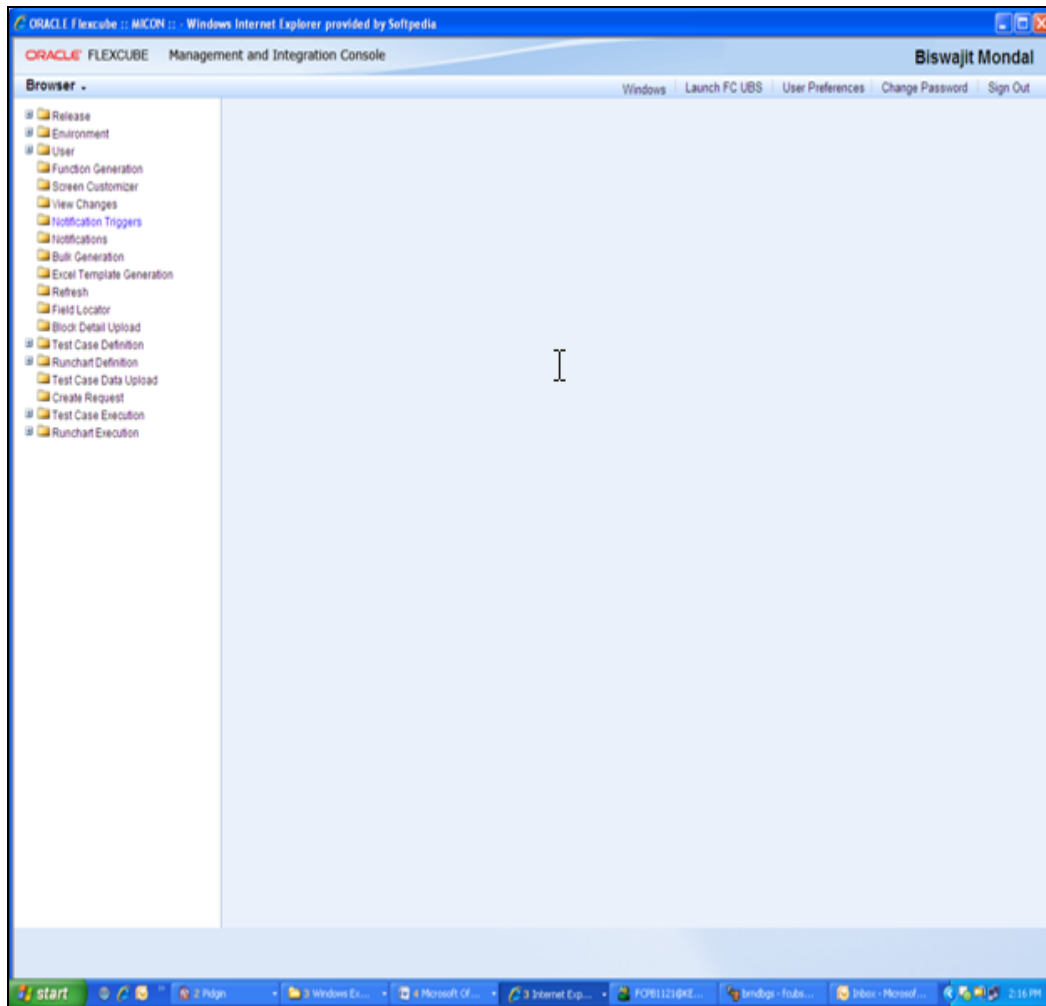
Example:

LDNTRG_LOG.txt

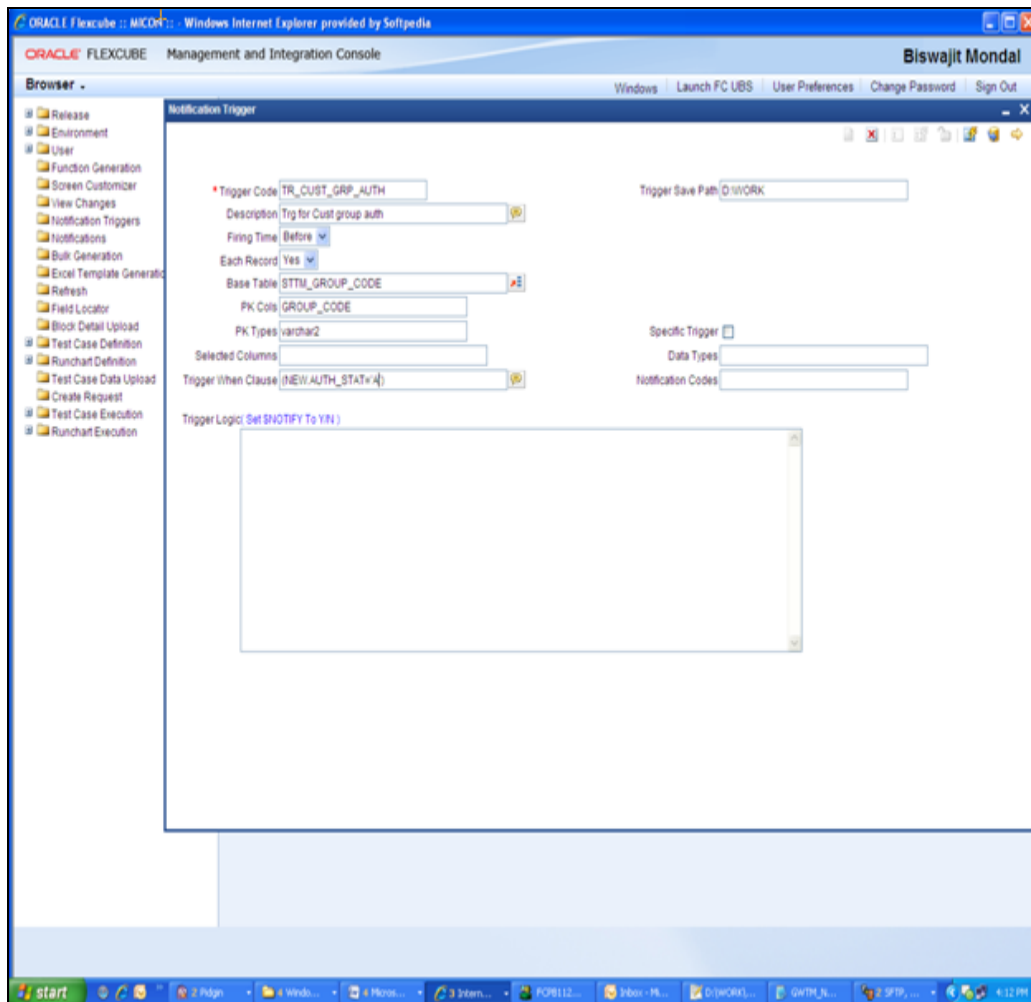
5.3 **Notification Trigger development**

5.3.1 **Development steps**

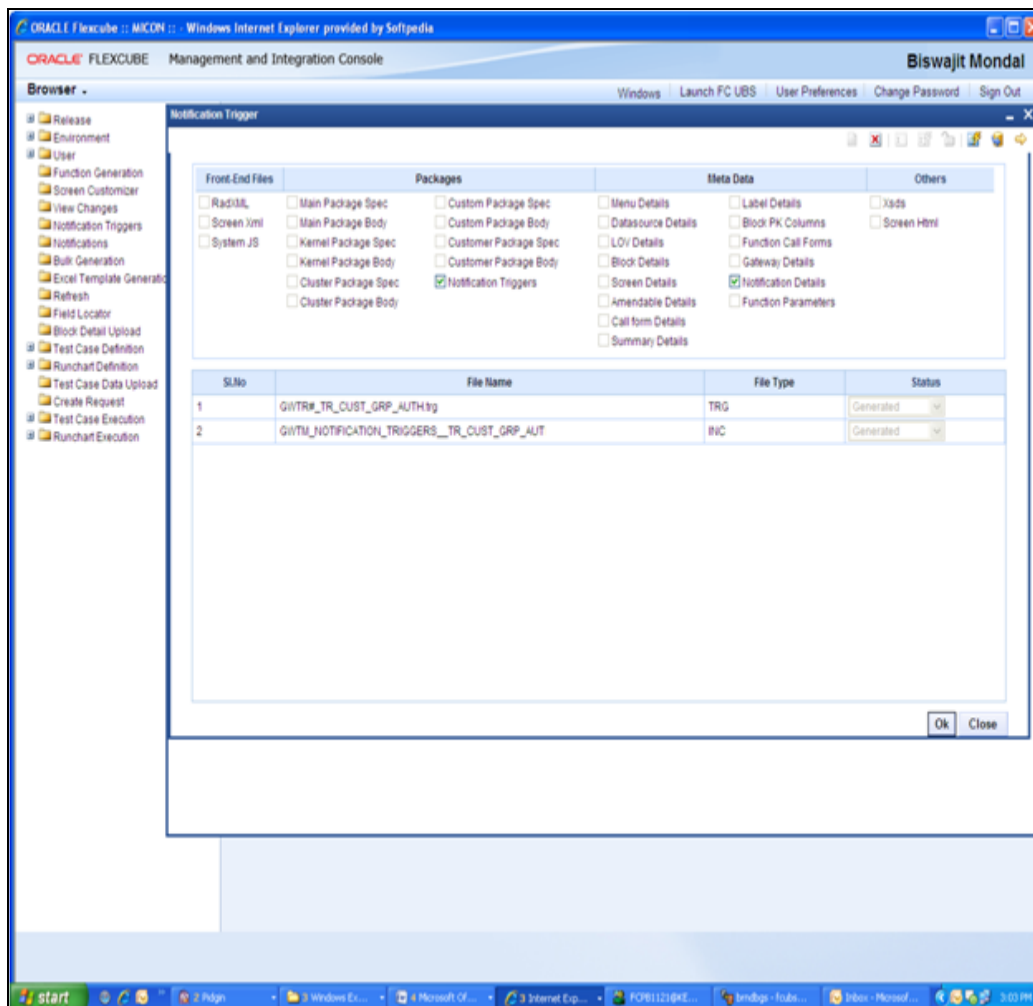
- Launch extensible Open Development and select “Notification Triggers”



- Select New and provide details as per the specification prepared earlier



- Save and generate files



5.3.2 Files

Notification Trigger development would create following files

- GWTM_NOTIFICATION_TRIGGERS__<Notification code>.INC
- GWTR#_<Notification code>.TRG

Example:

- GWTM_NOTIFICATION_TRIGGERS__TRIG_CONTRACT.INC
- GWTR#_TRIG_CONTRACT.TRG

5.4 Deploy Notification

5.4.1 Notification Open Development Related Deployment

Compile the following files in Target FLEXCUBE IS Database schema

- <module>pks_<Notification Function ID>_main.spc
- <module>pks_<Notification Function ID>_kernel.spc

- <module pks_<Notification Function ID>_custom.spc
- <module pks_<Notification Function ID>_cluster.spc
- <module pks__<Notification Function ID>__main.sql
- <module pks__<Notification Function ID>__kernel.sql
- <module pks__<Notification Function ID>__custom.sql
- <module pks__<Notification Function ID>__cluster.sql
- GWTM_NOTIFICATION_TAG_MAP__<Notification Function ID>_.INC
- GWTM_NOTIFICATIONS_MASTER__<Notification Function ID>_.INC
- VARIABLE_MAPPING__<Notification Function ID>_.INC

5.5 **Notification Trigger Deployment**

Compile the following files in Target FLEXCUBE IS Database schema

GWTM_NOTIFICATION_TRIGGERS__TRIG_CONTRACT.INC

GWTR#_TRIG_CONTRACT.TRG

5.6 **Test Notification**

This section explains the run time notification flow and testing steps

5.6.1 **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.

5.6.2 **Scheduler Based Notification**

1. The Notification Process in FLEXCUBE can be done using the jobs scheduler as follows:
2. The trigger generated from Open Development will be inserting key details into a static notification log (STTB_NOTIFICATION)
3. 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.
4. Each unprocessed record is locked.
5. The record is verified against the notification maintenance and checked whether notification is to be sent or not.
6. 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.
7. The job is then rescheduled to fire next time based on the previous execution.

Refer Gateway Installation documents on how to setup the Queues

5.6.3 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

5.6.4 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



Notification Development
[May] [2019]
Version 14.1.0.0.0

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], [2019], 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.