Patch Release Notes
Oracle Banking Payments

Release 14.7.4.0.0

June [2024]

ORACLE
Financial Services

Table of Contents

1. Release Notes

1.1 Background

Oracle Financial Services Software Services Limited has developed the Oracle Banking Payments, a stand-alone Payments Product Processor, to cater to the requirements of both Retail & Corporate segments. The agile and scalable nature of the solution helps Banks in quickly adapting to market changes. This is a Unified Payments platform for Local Clearing (or) Low Value, High Value (or) Large Value (RTGS), Cross-Border (SWIFT) and Book (or) Internal Transfer payments.

A patch release is an official Oracle patch release for Oracle Banking Payments. The Third, fourth & fifth digit in a patch release string indicates the patch release number. The first two digits indicate the release to which the patch is applied. Each patch release includes libraries and files that have been rebuilt to implement one or more fixes. Each patch release is incremental. In other words, the latest patch release includes new fixes released after last patch released for the same release.

1.2 Purpose

The purpose of this Release Note is to propagate the enhancements in Oracle Banking Payments 14.7.4.0.0.

1.3 Patch Requirements

Customers installing this release should refer to 14.7_Patchset_Deployment_Guidelines.docx included in the Patch Release zip file.

Confirm you are applying this patch release after applications of the previous patch releases up to 14.7.3.0.0 Patch set completed successfully.

1.4 Applying the Patch Release

Customers installing this patch release should refer to the "Installation Guide" chapter in the Oracle Banking Payments Release 14.7 Install and Upgrade Guide, which is available here:

https://docs.oracle.com/cd/F74675_01/index.htm

Customers installing this patch release should refer to the Readme included in the Patch Release zip file.

1.5 Release Highlights

The scope of the Oracle Banking Payments 14.7.4.0.0 Release is to develop new features apart from making enrichments to the existing functionality.

1.6 Issues Resolved

Issues Resolved.xlsx

Note: Click on the Attachment section of the PDF to access the attached excel sheet.

1.7 Extensibility Changes

NA

2. Enhancements

2.1 Generic Wire ISO

2.1.1 ACH CT - Date Derivation Changes

1. Changes are done to consider Requested Execution Date as either Instruction Date or Activation Date from requests received from channels /bulk file /UI.

For this a new field 'Requested Execution Date Considered as' is added in Source Network Preferences (Function ID: PMDSORNW). Options are Instruction Date (Default option/existing support) & Activation Date.

2. Debit Value date basis can be configured as Instruction Date (existing) or Activation Date. In Source Network preference (Function ID: PMDSORNW) new field is added as Debit Value Date Basis: This is a dropdown field with values Instruction Date (Default) and Activation Date.

Changes in Cross border MT and MX date derivation to check debit value date basis based on PMDSORNW. Debit Value date basis field is removed from PMDNWPRF and Generic wires preferences

2.1.2 Generic Wires ISO - Multiple RTGS Networks - XSD Validation / Queue Connectivity

- RTGS ISO -Changes are done to resolve Network based on MCS, Host Code, Network service ID & currency. When inbound RTGS ISO messages are received, system derives the Network based on the below logic
- MCS is derived based Queue/Folder in which message is received. Network is derived based on new maintenance Network Media Control System maintenance (Function ID: PMDNWMCS) available. If multiple Networks are derived, then Network Service ID and currency received in the message are used to derive the Network. If Network could not be derived, then transaction is moved to Repair queue•
- A new maintenance is provided to capture the XSDs for each message type against the Media Control System (Function ID: PSDGRXSP). XSD will be picked up from the newly introduced maintenance to validate against the XSD for the incoming RTGS ISO messages.

  Note: It is assumed that xsd for a message type received in a particular queue or folder linked to a Media Control system is same irrespective of the Network. If xsd is not same, messages are to be received in a separate queue or folder.

2.1.3 FI to FI Cancellation Request (camt.056) Additional Changes

Support for -

1. Making Assignment Identification field as an editable field on the screen
2. Defaulting Assignment Identification field with the value as NONREF on clicking Enrich user action and allowing user to modify the defaulted value

2.1.4 RTGS Directory Upload

Support for -

1. Capturing Network Directory Key during Target 2 RTGS Directory Upload through File envelope

2.1.5 Special Character Validations

Support for -

1. Performing Special character check for outbound camt.056, pacs.004, camt.029 messages
2. Replacing disallowed special character on Enrich user action with Replacement Character

2.2 Payments Core

2.2.1 Branch Code, Customer Name User ID, Account, Account Description - Length changes

Maximum allowed column length for the following entities is changed:

- Branch Code VARCHAR2(6)
- User ID, VARCHAR2(320)
- Customer Number VARCHAR2(20),
- Customer Name VARCHAR2(140),
- Account Number VARCHAR2(34)
- Account descriptionVARCHAR2(140) India RTGS

2.2.2 Separate indicators in ECA request for balance /account status override

The following changes are done for External Accounting systems with capability to process the transactions overriding Account status/balance related exceptions. Changes are done at ECA system maintenance, request and retry to record the details of the overridden exceptions at ECA level by user:

1. In External Credit Approval System screen-->Status mapping sub screen -->Reject category field is provided to indicate the type of the reject reason: Null (Default value), Error – No Retry, Account Balance Override, Account status Override, Status & Balance Override

   Note: For FCUBS co-deployed cases which returns only R status, reject category is to be maintained as one of the following values: Null, Status & Balance override (if force post is to be allowed), Error – No Retry

2. External Credit Approval Queue --> Retry Action screen (Function ID: PQDACQAU)

   'Skip Account Status Checks' Flag is added which is enabled enabled only for Reject category 'Account Status Override'.

   'Skip Account Balance Checks' Flag is enabled only if Reject category is 'Account Balance Override'

   Existing flag 'Retry with Force Post ' is enabled if Reject category is Status & Balance Override

   Note: It is external ECA system responsibility to decide whether to skip the validations which failed in the previous attempt or to skip all validations. There is no indication of the previous list of errors from Payments side.

   ECA Request generation changes based on retry screen flags:

   'Skip Account balance checks 'only is checked , in ECA request , FORCEBLOCK tag is populated as 'B' ;If the flag 'Skip Account Status Checks 'flag only is checked, FORCEBLOCK value is populated as Y ;If Retry with Force Post flag is checked , FORCEBLOCK value is populated as F (existing which is the FCUBS supported value)

   A new system parameter APPROVE_EAC_REJECTS is provided so that EAC records which are in Rejected status can be manually approved. If this parameter value is maintained as Y, then manual approval of EAC reject records will be allowed. If the parameter value is N or not maintained, approval from EAC screen is not allowed for rejected records.

   Note: If Accounting system has the capability to process the exceptions successfully, then only this parameter to be enabled.

2.2.3 Date re-derivation for Future Valued Batches / Transactions

If the Activation Date is in future and transaction is received in bulk file, consolidated batch is moved to Warehouse queue. When the consolidated Batch or transaction is released from the Warehouse queue, date re-derivation is not happening and the accounting and dispatch are processed with back date, if the Activation Date is marked as a holiday. Changes are applicable to ACH CT, Book Transfer and Cross border/RTGS MX

2.2.4 UETR Related Changes

UETR is made part of ECA/EAC/Accounting dto so that extensible changes can be done for external systems to consume this value. UETR mapping in the respective is applicable only to CBPR+/ RTGS ISO transactions for which UETR is available.

2.2.5 Batch Booking flag value addition

Batch Booking flag value is added in Browser Batch Booking Browser Detailed PMDBATBR and summary PMSBATCH.

2.2.6 Notification Generation Changes

- New screens Notification Preference (Function Id: PMDNOTIF /PMDBTNOT) are introduced to capture network code level, notification preference for the payment. For generation of notification, along with above maintenance “Notification Required” flag should be enabled for the source code in Source Maintenance Details (Function Id: PMDSORCE) screen.
- Notification generation can be configured for predefined set of payment statuses and queue statuses. The changes are applicable to ACH CT, Book Transfer & Generic wires for transaction level statuses/queue actions. In “System Parameters Detailed” screen, (function id: PMDSYSPM), new system parameter NEW_NOTIF_SERVICE_PAYMENT_TYPE, must be maintained with list of Payment Types separated by #, for which new notification service needs to be enabled. For example, value: Y#B#. Notification format is changed to include original value and enriched value; Notification for batch level queue actions; Notification on dispatch of messages at individual transaction level; Notification format is enhanced to include dispatched XML /inbound source XML messages for Generic wires.

2.2.7 ECA Auto Retry Changes

ECA Retry Preference (Function Id: PMDECAPR) is introduced to maintain the host code + network code + source code wise auto ECA retry preference. External Credit Approval System Detailed (Function Id: PMDECAMT) is modified to introduce new sub screen to capture "ECA Auto Retry Preference. The ECA response status is "Reject”, then based on the parameters set , system does auto retry .New timer job "AUTO_ECA_RETRY" is introduced to perform the Auto ECA retry .New auto job ECA Carry Forward (Function id: PMDECCFD) is introduced to carry forward the payment pending from ECA queue with status as Reject

2.2.8 Providing API for start & stop of jobs

New ReST service ‘PMJobStatusService’ is introduced to control the auto job status, Once the status is changed to Hold, API service should be invoked to change the status back to Active. This is applicable for auto jobs, seed jobs and timer jobs.

2.3 India UPI

2.3.1 India Payments - UPI API Changes

Bank APIs

1. Validate Address API – This API is initiated by PSPs and sent to the bank through NPCI whe1n PSP customer want to add a beneficiary within PSP application (for sending & collecting money).-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-nonfin-service/ReqValAdd/2.0/urn:txnid:<<txnId>>
2. Balance Enquiry API – This API allows user to enquire account balance of a user.-/obpm-india-upi-fin-service/ReqBalEnq/2.0/urn:txnid:<<txnId>>
3. Request Pay API – This API is used by NPCI to post Debit or Credit transactions to the accounts.-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-fin-service/ReqPay/2.0/urn:txnid:<<txnId>>
4. Check Transaction API – This API is used to request transaction status-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-fin-service/ReqChkTxn/2.0/urn:txnid:<<txnId>>

PSP APIs

1. List Account Service API – This API is used to find the list of accounts linked to the mobile number in particular bank.-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-nonfin-service/ReqListAccount/2.0/urn:txnid:<<txnId>>
2. OTP Service API – This API is used to request OTP for a particular customer account.-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-nonfin-service/ReqOtp/2.0/urn:txnid:<<txnId>>
3. Register Mobile Service API – This API is used to register upi account and set UPI PIN for the first time.-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-nonfin-service/ReqRegMob/2.0/urn:txnid:<<txnId>>
4. Balance Enquiry Service API – This API is used to enquire the account balance.-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-fin-service/ReqBalEnq/2.0/urn:txnid:<<txnId>>
5.Validate Address Service API – This API is initiated by PSPs and sent to NPCI when PSP customer want to inquire and validate beneficiary vpa.-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-nonfin-service/ReqValAdd/2.0/urn:txnid:<<txnId>>
6. Set Credentials Service API – This API is initiated by PSPs and sent to NPCI to set modified UPI PIN .-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-nonfin-service/ReqSetCre/2.0/urn:txnid:<<txnId>>
7. Heartbeat Service API – This API is the mechanism to monitoring the connection between PSP UPI switch and NPCI UPI switch-<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-fin-service/ReqHbt/2.0/urn:txnid:<<txnId>>
8. Transaction Confirmation Service API – This API is sent from NPCI to PSP for providing transaction confirmation -<<protocol>>://<<host>>:<<port>>/obpm-india-upi-psp-fin-service/ReqTxnConfirmation/2.0/urn:txnid:<<txnId>>

2.4 Cross Border

2.4.1 Inbound Charge Claim for gpi Payments

Support for

1. Skipping the underlying transaction gpi flag for Incoming Charge Claim (MT191) requests
2. Skipping the underlying transaction gpi check for Incoming Charge Claim (MT191) requests based on System Parameter

2.5 India IMPS

2.5.1 India Payments - IMPS MCC codes for Credit Card Transactions

MCC codes for Credit Card Transactions - Support is provided to populate Credit Card payment specific MCC codes - MCC 5415 (Corporate) or MCC 5414 (Retail) in IMPS outbound transactions

2.5.2 IMPS UDIR Changes

Request from NPCI to Remitter Bank to credit back the remitter (for reversal)Modified API | 'ReqPay' API initiated from NPCI to Remitter Bank to credit the remitter account online.

Response from Remitter Bank to NPCI to provide status of online credit (for reversal)Modified API | 'RespPay' initiated from Remitter Bank to NPCI to provide the status of the online credit transaction.

Screen Change | IMPS Outbound Transaction View Detailed (PJDOVIEW) -->
New button 'View Reversal Transaction' added

Screen Change | IMPS Inbound Message Browser (PJSINBRW)

I. New search criteria 'Transaction Sub-Type' with list of values –
   a. PAY
   b. DR_REVERSAL.

II. New button 'View Reversal Reject Details'.

Screen Change | IMPS Inbound Transaction View Detailed (PJDIVIEW),
new display fields added –

i. 'Original Retrieval Reference'
ii. 'Transaction Sub-Type'

Screen Change | IMPS Inbound Transaction View Summary (PJSIVIEW)

I. New search criteria 'Transaction Sub-Type' with list of values –
   a. PAY
   b. DR_REVERSAL

New Screen | Reject Reversal Details (Function ID: - PJDREVDL) – To display the reject details of remitter online credit transaction if the original matching transaction is not

found

2.6 India NEFT

2.6.1 ISO Changes

- NEFT migration from IFN298 format to ISO20222 format -
- New System Parameter defined to support NEFT IFN Messages and ISO Messages.
- XSD validation supported for outgoing and incoming NEFT ISO messages.
- Additional fields for NEFT ISO added to PTDOTONL, PTDOVIEW, PTDITONL, PTDIVIEW, PTSOVIEW, PTSIVIEW screen.
- NEFT migration from IFN298 format to ISO20222 format
- New System Parameter defined to support NEFT IFN Messages and ISO Messages.
- XSD validation supported for outgoing and incoming NEFT ISO messages.
- Additional fields for NEFT ISO added to PTDOTONL, PTDOVIEW, PTDITONL, PTDIVIEW, PTSOVIEW, PTSIVIEW screen.
- Screen changes done to PTSOUTBR and PTSINBRW Browser screens.
- N03 browser Screen/field Name changes to align with ISO terminology.
- N04 Browser/Manual input of N04 screen name changes to align with ISO terminology.
- Processing and bulking of outgoing NEFT payments initiated through UI, Channel/SPS and generate pacs.008.001.09 (formerly N06) messages.
- Generation and bulking of pacs.004.001.10 (formerly N07) messages, camt.059.001.06 (formerly N10) credit confirmation messages.
- Generation of Business Application Header (formerly Block A) as per specification for every message sent.
- Parsing and de-bulking of incoming NEFT payments which are uploaded as pacs.008.001.09 (formerly N02) messages.
- Processing of camt.059.001.06 (formerly N10) credit confirmation messages, admi.004.001.02 (formerly F Series) ACK/NAK messages, Pacs.002.001.11 reject message (formerly IFN298N03), camt.052.001.08 (formerly IFN972) SOD message.
- Service Changes done for Validation Service, SPS service to add/map LEI and Category Purpose Code tags.
- Pain.001 mapping changes for LEI fields, category purpose code
- FCRA Validation for NEFT outbound

2.7 EU SEPA

2.7.1 SCT/SDD auto reject changes

Currently if DDA system returns multiple error codes, system checks for the presence of the error code linkage in PMDRJMNT screen for the combined error code list with error codes

separated by semicolon. This is changed to check for individual error codes and if any error code linkage is present and if auto cancel is enabled for reject status, auto return is processed.

2.8 EU SEPA Instant

2.8.1 Routing matrix support SEPA Instant

Routing matrix support is extended for SEPA Instant for UI, services and bulk file upload.

2.8.2 pain.002 generation

pain.002.001.10 generation is supported for records uploaded by pain.001.001.09.

2.9 Book Transfer

2.9.1 Accounting Branch input is allowed for GLs in Book Transfer

Accounting Branch input is allowed for GLs in Book Transfer. Changes are done in singlepayout service to received account branch field (optional).

2.10 Core

2.10.1 Technical Message ID duplicate handling changes

When duplicate Message ID is received in channel requests, system picks up the response status and response from PMTB_OUTGOING_LOG for the Original Transaction Id and send the same response. If duplicate message id is received more than 10 times Error Response will be sent as: Too many request with same Message ID already received, hence ignored (PM-MSG-006). The number of times duplicate message ID is to be received for generation of the error message is made configurable using system parameter MSGID_DUP_ORGNL_RESP_COUNT.

2.10.2 Additional Rest services for OBDX

Additional Rest services are provided for

SEPA Directory query
http://<host>:<port>/PMReST/obpmrest/payments/SEPADirectoryQuery
Outbound records received in MT 101 --> To list MT 101 records
http://<host>:<port>/PMReST/obpmrest/payments/queryMT101List
To get details of a MT 101 record
http://<host>:<port>/PMReST/obpmrest/payments/getMT101Details

2.10.3 Bulk retry support for ECA/EAC queues

Support is provided for selecting bulk records when manual retry action is initiated from ECA/EAC queue to FCUBS.

2.10.4 ReST multi part API integration with OBDX for C2B files

This Rest API is to initiate the file envelope upload. File is sent as a multipart/form-data in the request along with the file envelope details. This API uploads the file to the server in the configured path
and then processes the file envelope .http://<host>:<port>/PMReST/obpmfile/fileenvelop

2.11 Bulk File

2.11.1 Bulk File with Consolidated debit for Cross border/RTGS MT

1. If the processing type for Cross border /RTGS MT is maintained as N in the static table PMTB_PAYMENT_PREFERENCE_TYPE, then it will be possible to process consolidation of debits similar to other non-urgent payments regrouping the records based on Instruction Date, Activation Date, Network, Transfer Currency, CO ID & FX reference.
2. Non urgent preferences maintained in PMDONPRF/PMDONCST are applicable and customer file preference PMDFLPRF is to be maintained. It is possible to configure either itemized or consolidated posting . Batch preference PMDBTPRF and Batch validation Preference PMDBTVAL are to be maintained.
3. For itemized accounting and for credit accounting in case of consolidated cases, sequencing of accounting is done when request is sent to FCUBS. Number of accounting entries which can be sent in an accounting request can be configured in system parameter MAX_EXT_TXN_ACCOUNTING_LIMIT.
4. Network cutoff check and messaging are done only on successful completion of accounting.

3. Deprecated Features

NA

4. Components of the Software

- For information on the Components of the Software, please refer Oracle Banking Payments 14.7.0.0.0 release notes section 4.

5. Tech Stack

- For information on the Technical stack, please refer Oracle Banking Payments 14.7.0.0.0 release notes section 5.

6. Third Party Software Details

- For information on the third-party software details, please refer Oracle Banking Payments 14.7.0.0.0 License Guide.

7. Non-Functional Enhancements

7.1 Dynamic Value generation for server.id

- Application instance will get a unique value allocated for server.id internally.
- On a graceful shut down of an application instance, server.id value allocated for this instance will return to pool.
- In case of a crashed instance, the server.id value will not go back to pool immediately. A monitoring job will check and recover this value back to pool. This will regulate the continuous growth of values for server.id
- The values generated for server.id is in the range of 1 to 99
- The value for server.id allocated from a pool is not sticky to a node. For example, in a cluster of 5 nodes, if node1 gets server.id values allocated as 1 and node2 gets server.id value allocated as 2; on a complete restart of all the nodes, the previous allocated number may shuffle across the nodes. Coincidentally each node may get back the same value upon restart, but it is not guaranteed.
- The allocation and de-allocation of the server.id values are tracked centrally in a database.
- Possibility of race-condition is anticipated when all the nodes can start together and this situation is taken care to avoid allocation of duplicate value to multiple nodes

  Please refer to section 7.3 for system configuration changes.

7.2 Master Jobs High Availability

- This feature provides a self-resiliency to the master jobs in the application against the failure of a master node on which these master jobs are running.
- One of the active application instances will get designated as a master and starts the master jobs on its node.
- On a graceful shutdown of a master node, it de-registers itself as a master and allows one of the other active application instances to take over as a master and the new master node will start the master jobs.
- On a crash scenario, a health check job running on all the nodes will designate one of the application instance as a master. This new master will start the master jobs.
- In either of a scenario (graceful shutdown or crash), the master node switch-over will happen only during the next run of a health check job, so a little delay in switch-over can be expected Application instance will get a unique value allocated for server.id internally. On a graceful shut down of an application

Please refer to section 7.3 for system configuration changes.

7.3 System Configuration Changes

7.3.1 De-supported Configurations

Managed Server / JVM Startup parameter, -Dserver.id:
This parameter is no longer supported by the application. Even if this parameter is set as part of JVM start-up argument, this will be ignored.

Function Id: PMDSYSPM
Menu Path: Payments Maintenance -> Common -> System Parameters Detailed

Parameter Key: BATCH_SERVER_ID

This key is used to designate one of the nodes as a master node. A value for this key is -Dserver.id value of one of the nodes in a cluster. A master node is responsible to start the master jobs within it. If the master node is shutdown / crashes, manually this parameter value should be updated with another active node's -Dserver.id value and restart the new master node, to start the master jobs.

7.3.2 New Configurations

- ADD BEFORE APPLYING PS - WEBLOGIC CONSOLE: New DataSource creation (with Supports Global Transaction and LLR (Last Logging Required enabled))
  Name of Datasource = jdbc/fcjdevHADS_GTXN).

- REMOVE BEFORE APPLYING PS - WEBLOGIC CONSOLE: Remove the Server Startup Argument(-Dserver.id = 1)

- COMPILE BEFORE APPLYING PS - DB: All released DDL INCs related to tables PMZB_LEADER_INSTANCE_REGISTRY, PMZB_INSTANCE_REGISTRY to be compiled in App Root Schema along with CSZM_APPROOT_OBJECTS.

  Factory Shipped Configuration Data with default values:

  Configurations related to these new feature are released as INC for table PMZM_INSTANCE_PROPERTIES.

  It is suggested to run this feature with the default values. If any issue is noticed running with these default value, then please consult the implementation partner or engineering team based on the support access

- Below properties are released into Config Service with the default values.

Key	Description	Value
`dynamic.instance.id.retry.count`	Total number of retry attempts to generate the server.id value. In case of cluster targeted deployment, many instances will attempt to generate the id and there may be clash. In such scenario, one application instance can succeed to claim the id and rest of the instances will attempt to generate new id. After a maximum attempt, an application instance will terminate the deployment. For a cluster with nodes higher than 3, this number can be adjusted suitably.	3
`dynamic.instance.id.active.interval`	The time duration beyond which if the health of a instance id is not refreshed, that instance id will be treated as stale and a corresponding health check job will claim such id back in to the available id pool.	17 (in minutes)
`instance.liveness.timeout`	The instance id health check job will probe the liveliness of an instance through HTTP GET. This is in addition to the `dynamic.instance.id.active.interval` to ensure that the instance is actually not responding. This parameter defines the timeout for HTTP call.	1 (in minute)
`instance.liveness.retry.count`	Number of attempts for the liveliness probe in case of any error.	2
`instance.health.job.run.interval`	Interval between the runs of Instance Health Monitor Job.	15 (in minutes)
`leader.retry.count`	Total number of retry attempts to elect a leader instance. In a clustered deployment, multiple instances may claim a leader position, but only one can succeed. This retry will ensure that the claim failure is authentic and also other instances will ensure that one instance is elected as a leader.	3

	If an application instance crossed this retry count, this means no instance has been elected as a leader and the deployment will terminate.	
`leader.health.job.run.interval`	Interval between the runs of Leader Health Monitor Job.	6 (in minutes)

Patch Release Note
June [2024]
Version 14.7.4.0.0

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], [2024], 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.