

Oracle® Retail Merchandising System

Installation Guide

Release 15.0.2

E90685-03

July 2018

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Primary Author: Preethi Sahu

Contributors: Nathan Young

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs 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 fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

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

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

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via**™ licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex**™ licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments	xi
Preface	xiii
Audience	xiii
Customer Support.....	xiii
Review Patch Documentation	xiii
Improved Process for Oracle Retail Documentation Corrections	xiii
Oracle Retail Documentation on the Oracle Technology Network.....	xiv
Conventions.....	xiv
1 Preinstallation Tasks	1
Installation Terminology	1
Implementation Capacity Planning.....	1
Requesting Infrastructure Software.....	2
Check Supported Database Server Requirements.....	2
Check Supported Application Server Requirements	3
Verify Single Sign-On.....	4
Check Supported Web Browser and Client Requirements	4
Supported Oracle Retail Products	4
Supported Oracle Retail Integration Technologies	5
Supported Oracle Applications.....	5
UNIX User Account Privileges to Install the Software	5
Verify RMS and SIM Inventory Adjustment Reason Codes	5
Data Access Schema (DAS).....	6
2 RAC and Clustering	7
Part I: Full Installation	9
3 Database Installation Tasks—Full	11
Data Access Schema	11
RMS Database Schema Distribution – Oracle Retail Applications Included	11
Create Staging Directory for RMS Installer	11
Establish a Database Partitioning Strategy	12
Step 1: Modify partition_attributes.cfg	14
Step 2: Modify Data Definition Files	14
Step 3: Generate DDL for DAS Tables – Run partition.ksh (Optional)	15
Step 4: Generate DDL for Tables – Run partition.ksh	16
Create the RMS Database.....	17
Create the Database Instance Using Oracle Generic Template	17
Create Required RMS Tablespaces.....	18
Create the Schema Owner for RMS.....	19
Create the Database User RMS_ASYNC_USER	20
Create the Database User for Allocation (Optional)	20

Create the Database User for Demo Data (Optional).....	20
Create the Database User for DAS (Optional)	21
Run the RMS Database Schema Installation	21
Values to Remember for the Batch and Application Installation.....	23
Resolving Errors Encountered During Database Schema Installation	23
Set Up Additional RMS Users.....	23
PRODUCT_VERS_CONFIG_OPTIONS	24
Batch Security Setup	24
Adding a User to the RPM Application.....	25
4 Batch Installation Tasks—Full.....	27
Create Staging Directory for RMS Installer	27
Run the RMS Installer.....	28
Resolving Errors Encountered During Batch Installation	29
Manual Steps for Running script ld_iindfiles.ksh	29
RETL	30
Data Conversion Scripts.....	30
5 Application Server Installation Tasks – Full	31
Prepare Application Server for RMS.....	31
Create RMS Help Managed Server.....	31
Install NodeManager.....	34
Create Staging Directory for RMS Application Server Files	37
Run the RMS Application Installation	38
Verifying FORMS Configuration File Details	40
Resolving Errors Encountered During Application Installation.....	41
Webutil Setup	41
Clustered Installations – Post-Installation Steps.....	43
RMS Reports Copied by the Application Installation.....	43
6 Oracle BI EE Configuration for RMS Reports	45
BI Server Component Installation Tasks.....	45
Installation Process Overview.....	45
Post Install Steps for OBIEE 12C.....	46
Installing the RMS BI Publisher Templates.....	51
Configuring the RMS JDBC Connection.....	52
Part II: Upgrade Installation	55
7 Database Installation Tasks – Upgrade	57
Upgrade RMS Database using the Installer.....	57
Create Staging Directory for RMS Database Schema Files	58
Optional- Analyze Changes in the Patch.....	58
Run the Allocation Data Cleanup Scripts.....	59
Run the RMS Database Schema Upgrade.....	59
Resolving Errors Encountered During Database Schema Installation.....	60

8 Batch Installation Tasks—Upgrade	63
Create Staging Directory for RMS Installer	63
(Optional) Analyze Changes in the Patch.....	64
Run the RMS Installer.....	64
Resolving Errors Encountered During Batch Installation	65
Manual Steps for Running script ld_iindfiles.ksh	66
9 Application Server Installation Tasks – Upgrade	67
Create Staging Directory for RMS Installer	67
(Optional) Analyze Changes in the Patch.....	67
Run the RMS Application Installation	67
10 Reports Installation Tasks – Upgrade	71
Installing the RMS BI Publisher Templates	71
11 Data Migration	73
Create Staging Directory for RMS Data Migration Files	73
Configure RMS Data Migration Tool	73
Run the RMS Data Migration Tool	75
Run the ReIM Data Migration.....	76
Run the Allocation Data Migration	76
12 Web Services Installation	77
Set up Environment.....	77
Grant permissions to RMS Database Schema	78
Create a Managed Server.....	78
Create a Datasource.....	78
Deploy RMS Service EAR File.....	78
Configure Web Service Security	80
Configuring RMS RESTful Web Services:	81
Create a Managed Server.....	81
Create a Datasource.....	81
Deploy RMS RESTful Web Service File	81
Configure Users and Group:	82
13 Patching Procedures	83
Oracle Retail Patching Process	83
Supported Products and Technologies	83
Patch Concepts	84
Patching Utility Overview	85
Changes with 15.0.....	85
Patching Considerations	85
Patch Types.....	85
Incremental Patch Structure	86
Version Tracking.....	86
Apply all Patches with Installer or ORPatch.....	86
Environment Configuration	87

Retained Installation Files.....	87
Reloading Content.....	87
Java Hotfixes and Cumulative Patches.....	87
Backups.....	88
Disk Space.....	88
Patching Operations.....	89
Running ORPatch.....	89
Merging Patches.....	99
Compiling Application Components.....	100
Deploying Application Components.....	102
Maintenance Considerations.....	103
Database Password Changes.....	103
WebLogic Password Changes.....	104
Infrastructure Directory Changes.....	105
DBManifest Table.....	105
RETAIL_HOME relationship to Database and Application Server.....	105
Jar Signing Configuration Maintenance.....	105
Customization.....	107
Patching Considerations with Customized Files and Objects.....	107
Registering Customized Files.....	108
Custom Compiled Java Code.....	110
Extending Oracle Retail Patch Assistant with Custom Hooks.....	112
Troubleshooting Patching.....	117
ORPatch Log Files.....	117
Restarting ORPatch.....	117
Manual DBManifest Updates.....	117
Manual Restart State File Updates.....	119
DISPLAY Settings When Compiling Forms.....	119
JAVA_HOME Setting.....	119
Patching Prior to First Install.....	119
Providing Metadata to Oracle Support.....	120
A Appendix: Oracle 12cR1 Database Parameter File.....	123
B Appendix: Configure Listener for External Procedures.....	125
C Appendix: Tablespace Creation.....	127
Non-Encrypted Tablespace Creation.....	127
Encrypted Tablespace Creation.....	127
Configure a Wallet.....	127
Encryption at Tablespace Level.....	128
D Appendix: RMS RETL Instructions.....	129
Configuration: RETL.....	129
E Appendix: Oracle Trade Management System Expectations.....	133
Installation Scripts (elc_comp_post_htsupld.sql).....	133

HTS Upload / Mass Update.....	135
Calculation of Merchandise Processing Fee.....	136
Unit of Measure Conversions.....	136
Customs Entry Ref. Status	136
Customs Entry Totals	137
F Appendix: RMS Database Schema and Batch Installation Screens.....	139
G Appendix: RMS Application Installer Screens.....	173
H Appendix: RMS Analyze Tool.....	185
Run the RMS Analyze Tool	185
I Appendix: Installer Silent Mode	187
J Appendix: URL Reference	189
JDBC URL for a Database	189
K Appendix: Common Installation Errors.....	190
RMS Installer unable to connect to the database	190
Database Installer Hangs on Startup.....	191
Warning: Could Not Find X Input Context.....	191
Unresponsive Country and Currency Drop-Downs.....	192
Could Not Execl Robot Child Process: Permission Denied.....	193
ConcurrentModificationException in Installer GUI.....	193
FRM-30064: Unable to Parse Statement Select While Compiling fm_ituda.fmb	193
ORA-04031 (Unable to Allocate Memory) Error During Database Schema Installation	194
X Error of Failed Request: BadWindow (Invalid Window Parameter)	194
RIB Errors.....	194
Error Connecting to Database URL.....	195
Multi-Threaded OCI Client Dumps Core after Reconnecting To Database	195
Forms Installer Fails on HP-UX	196
FRM -93552: cannot connect to runtime process. Error when using RMS in an SSO environment.....	196
Symptom	196
RUNNING 002065_retail_ctx.sql Creating Context 'RETAIL_CTX' CREATE CONTEXT RETAIL_CTX USING retail_ctx_pkg * ERROR at line 1: ORA-00955: name is already used by an existing object	197
FRM-30085: Unable to adjust form for output	197
L Appendix: Single Sign-On for WebLogic	199
What Do I Need for Single Sign-On?	199
Can Oracle Access Manager Work with Other SSO Implementations?	199
Oracle Single Sign-on Terms and Definitions	200
What Single Sign-On is not.....	201
How Oracle Single Sign-On Works	201
Installation Overview	203
User Management.....	203

M Appendix: Single Sign-On Resource Access Descriptors	205
N Appendix: AIX Shared Library Bug Fix.....	207
O Appendix: Inserting New Languages.....	209
Insert Primary Language Data	209
P Appendix: Setting Up Password Stores with wallets/credential stores.....	211
About Database Password Stores and Oracle Wallet	211
Setting Up Password Stores for Database User Accounts.....	212
Setting up Wallets for Database User Accounts	213
For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI	213
Setting up RETL Wallets	215
For Java Applications (SIM, ReIM, RPM, RIB, AIP, Alloc, ReSA, RETL).....	216
How does the Wallet Relate to the Application?	219
How does the Wallet Relate to Java Batch Program use?.....	219
Database Credential Store Administration.....	219
Managing Credentials with WSLT/OPSS Scripts	221
listCred	222
updateCred	223
createCred	224
deleteCred.....	224
modifyBootStrapCredential	225
addBootStrapCredential	225
Quick Guide for Retail Password Stores (db wallet, java wallet, DB credential stores)	227
Q Appendix: Creating User Synonyms	237
R Appendix: Manual Forms Compilation	239
S Appendix: Manual Batch Compilation	241
T Appendix: Installation Order	243
Enterprise Installation Order.....	243

Send Us Your Comments

Oracle Retail Merchandising System, Installation Guide, Release 15.0.2

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com
Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Oracle Retail Installation Guides contain the requirements and procedures that are necessary for the retailer to install Oracle Retail products.

Audience

This Installation Guide is written for the following audiences:

- Database administrators (DBA)
- System analysts and designers
- Integrators and implementation staff

Customer Support

- To contact Oracle Customer Support, access My Oracle Support at the following URL:
 - <https://support.oracle.com>
- When contacting Customer Support, please provide the following:
 - Product version and program/module name
 - Functional and technical description of the problem (include business impact)
 - Detailed step-by-step instructions to re-create
 - Exact error message received
 - Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 15.0) or a later patch release (for example, 15.0.2). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times **not** be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part

number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement “the Window Name window opens.”

This is a code sample

It is used to display examples of code

Preinstallation Tasks

This chapter includes tasks to complete before installation.

Note: Oracle Retail assumes that the retailer has applied all required fixes for supported compatible technologies.

Installation Terminology

STAGING_DIR – The directory where the rms15installer.zip is copied and extracted locally.

RETAIL_HOME – The directory where Database Files are stored, and Batch and Forms are installed. This will contain the orpatch directory..

- Database RETAIL_HOME – The location where RMS Database Files are stored. This location will be used during the subsequent patching of the RMS.
- Batch RETAIL_HOME – This is the Batch installation directory, the location where RMS Batch Files are installed.
- Forms RETAIL_HOME – This is the Forms installation directory, the location where RMS Forms are installed.

Note: The RETAIL_HOME for database and batch can be the same. A separate RETAIL_HOME is required for forms installation.

Implementation Capacity Planning

There is significant complexity involved in the deployment of Oracle Retail applications, and capacity planning is site specific. Oracle Retail strongly suggests that before installation or implementation you engage your integrator (such as the Oracle Retail Consulting team) and hardware vendor to request a disk sizing and capacity planning effort.

Sizing estimates are based on a number of factors, including the following:

- Workload and peak concurrent users and batch transactions
- Hardware configuration and parameters
- Data sparsity
- Application features utilized
- Length of time history is retained

Additional considerations during this process include your high availability needs as well as your backup and recovery methods.

Requesting Infrastructure Software

If you are unable to find the necessary version of the required Oracle infrastructure software (database server, application server, WebLogic, etc.) on the Oracle Software Delivery Cloud, you should file a non-technical 'Contact Us' Service Request (SR) and request access to the media. For instructions on filing a non-technical SR, see My Oracle Support Note 1071023.1 – *Requesting Physical Shipment or Download URL for Software Media*.

Check Supported Database Server Requirements

General requirements for a database server running RMS include the following.

Supported on:	Versions Supported:
Database Server OS	OS certified with Oracle Database 12cR1 Enterprise Edition. Options are: <ul style="list-style-type: none"> ▪ Oracle Linux 6 and 7 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 and 7 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Database Server 12cR1	Oracle Database Enterprise Edition 12cR1 (12.1.0.2) with the following specifications: Components: <ul style="list-style-type: none"> ▪ Oracle Partitioning ▪ Examples CD Oneoffs: <ul style="list-style-type: none"> ▪ 20846438: ORA-600 [KKPAPXFORMFKK2KEY_1] WITH LIST PARTITION ▪ 19623450: MISSING JAVA CLASSES AFTER UPGRADE TO JDK 7 ▪ 20406840: PROC 12.1.0.2 THROWS ORA-600 [17998] WHEN PRECOMPILING BY 'OTHER' USER ▪ 20925154: ORA-39126: WORKER UNEXPECTED FATAL ERROR IN KUPW\$WORKER GATHER_PARSE_ITEMS JAVA RAC only: <ul style="list-style-type: none"> ▪ 21260431: APPSST 12C : GETTING ORA-4031 AFTER 12C UPGRADE ▪ 21373473: INSTANCE TERMINATED AS LMD0 AND LMD2 HUNG FOR MORE THAN 70 SECS Other components: <ul style="list-style-type: none"> ▪ Perl interpreter 5.0 or later ▪ X-Windows interface ▪ JDK 1.7

Note: By default, JDK is at 1.6. After installing the 12.1.0.2 binary, apply patch 19623450. Follow the instructions on Oracle Database Java Developer's Guide 12c Release 1 to upgrade JDK to 1.7. The Guide is available at: <http://docs.oracle.com/database/121/JJDEV/chone.htm#JJDEV01000>.

Check Supported Application Server Requirements

General requirements for an application server capable of running RMS include the following.

Supported on	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 12c (12.2.1.2). Options are: <ul style="list-style-type: none"> ▪ Oracle Linux 6 and 7 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 and 7 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11.3 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Application Server	Oracle Fusion Middleware 12c (12.2.1.2) Components: <ul style="list-style-type: none"> ▪ Oracle WebLogic Server 12c (12.2.1.2) ▪ Oracle Forms Services 12c (12.2.1.2) Java: <ul style="list-style-type: none"> • JDK 1.8+ 64 bit Other Components: <ul style="list-style-type: none"> ▪ Oracle BI Publisher 12c (12.2.1.2) ▪ Optional(Required for SSO) ▪ Oracle WebTier 12c (12.2.1.2) ▪ Oracle Access Manager 11g Release 2 PS3 (11.1.2.3) ▪ Oracle Access Manager Agent (WebGate) 11g Release2 PS3(11.1.2.3)

Verify Single Sign-On

If RMS is not being deployed in a Single Sign-On environment, skip this section.

If Single Sign-On is to be used, verify the Oracle Identity Management 11gR1 version 11.1.1.9 has been installed along with the components listed in the above Application Server requirements section. Verify the HTTP Server is registered with the Oracle Access Manager (OAM) 11gR2 PS3 as a partner application.

Check Supported Web Browser and Client Requirements

General requirements for client running RMS include:

Requirement	Version
Operating system	Windows 7 or 10
Display resolution	1024x768 or higher
Processor	2.6GHz or higher
Memory	1GByte or higher
Networking	intranet with at least 10Mbps data rate
Oracle (Sun) Java Runtime Environment	1.8+
Browser	Microsoft Internet Explorer 11 Mozilla Firefox Extended Support Release 52+

Supported Oracle Retail Products

Product	Version
Oracle Retail Insights	15.0.2
Oracle Retail Data Extractor for Merchandising (RDE)	15.0.2
Oracle Retail Active Retail Intelligence (ARI)	15.0
Oracle Retail Price Management (RPM)	15.0.2
Oracle Retail Allocation	15.0.2
Oracle Retail Invoice Matching (ReIM)	15.0.2
Oracle Retail Store Inventory Management (SIM)	15.0.2
Oracle Retail Warehouse Management System (RWMS)	15.0.2
Oracle Retail Advanced Inventory Planning (AIP)	15.0.2
Oracle Retail Merchandise Financial Planning (MFP)	15.0.2
Oracle Retail Demand Forecasting (RDF) (including the Grade module)	15.0.2
Oracle Retail Predictive Application Server (RPAS)	15.0.2
Oracle Retail POS Suite with Mobile Point-of-Service	14.1
Oracle Retail Xstore Suite	15.0.1

Supported Oracle Retail Integration Technologies

Integration Technology	Version
Oracle Retail Extract, Transform and Load (RETL)	13.2.8.0.1
Oracle Retail Integration Bus (RIB)	15.0.2
Oracle Retail Service Backbone (RSB)	15.0.2

Supported Oracle Applications

Requirement	Version
Oracle E-Business Suite Financials	Oracle E-Business Suite 12.2.4 integration is supported using the Oracle Retail Financial Integration for Oracle Retail Merchandising Suite and Oracle E-Business Suite Financials. See the <i>Oracle® Retail Financial E-Business Suite Integration Solution Implementation/Operations Guide</i> for specific version information.
Oracle PeopleSoft Financials	Oracle PeopleSoft Financials 9.2, integration is supported using the Oracle Retail Financial Integration for Oracle Retail Merchandising Suite and Oracle PeopleSoft Financials. See the <i>Oracle Retail Financial Integration for Oracle Retail Merchandise Operations Management and Oracle E-Business Suite or PeopleSoft Financials</i> for specific version information.

UNIX User Account Privileges to Install the Software

A UNIX user account is needed to install the software. The UNIX user that is used to install the software should have write access to the WebLogic server installation files.

For example, "oretail."

Note: Installation steps will fail when trying to modify files under the WebLogic installation, unless the user has write access.

Verify RMS and SIM Inventory Adjustment Reason Codes

SIM and RMS must have the same inventory adjustment reason codes to work properly.

Data Access Schema (DAS)

Data Access Schema (DAS) is an optional component of the Merchandising Suite. DAS exposes a subset of core RMS data to external applications via database replication. DAS allows these applications read-only access RMS data as they need it. The use of a separate schema on a separate database insulates core RMS processes from outside requests for information.

DAS includes a number of replicated foundation and inventory objects. The structure of these objects is identical to RMS. Additionally, DAS includes two layers of database views that help shape RMS data so it is more understandable to system integrators and 3rd party systems.

The default configuration discussed in Oracle Retail documentation describes using Oracle Streams to replicate data. Oracle Streams is included in the Enterprise Edition database license that all RMS customers have; and as of 12C, it is supported on a non-multi-tenant environment.

The main requirement of the solution is that data be replicated. Customer can use any preferred replication technology (for example, [Oracle GoldenGate](#)) that supports basic unidirectional replication in a container and/or a non-container environments.

RAC and Clustering

The Oracle Retail Merchandising has been validated to run in two configurations on Linux:

- Standalone WebLogic and Database installations
- Real Application Cluster Database and WebLogic Clustering

The Oracle Retail products have been validated against a 12.1.0.2 RAC database. When using a RAC database, all JDBC connections should be configured to use THIN connections rather than OCI connections. Forms connections will continue to use OCI connections.

Clustering for WebLogic Server 10.3.6 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 12.1.0.2 Oracle Internet Directory database with the WebLogic 10.3.6 cluster. It is suggested that a Web Tier 11.1.1.7 installation be configured to reflect all application server installations if SSO will be utilized.

References for Configuration:

- Oracle Fusion Middleware High Availability Guide 11g Release 1 (11.1.1) Part Number E10106-09
- Oracle Real Application Clusters Administration and Deployment Guide 12c Release 1 (12.1) E48838-10

Part I: Full Installation

Part I of this guide details the steps needed to perform a full installation of RMS. Part I contains the following chapters:

- [Database Installation Tasks—Full](#)
- [Batch Installation Tasks—Full](#)
- [Application Server Installation Tasks—Full](#)
- [Reports Installation Tasks —Full](#)

For information about an upgrade installation, see Part II.

Database Installation Tasks—Full

This chapter describes the tasks required for a full database installation.

Note: If the RMS 15.0.1 software is already installed, please see [Database Installation Tasks—Upgrade](#) for information on Upgrading to RMS 15.0.2.

Data Access Schema

Data Access Schema (DAS) exposes a subset of core RMS data to external applications via database replication. DAS allows these applications read only access RMS data as they need it. The use of a separate schema insulates core RMS processes from outside requests for information. The default configuration discussed in Oracle Retail documentation described using Oracle Streams to replicate data. Oracle Streams is supported on a non-multitenant environment only. To replicate in a multitenant environment, one may consider using other utilities such as Golden Gate.

If you choose to implement the DAS schema, execute the DDL scripts included in the upcoming sections.

RMS Database Schema Distribution – Oracle Retail Applications Included

The RMS 15.0.2 release contains an installer package that can be used to install the database objects for the following products: RMS, ReSA, RTM, RPM, ReIM, and Allocation.

Note: The Java application installers for RPM, ReIM, ReSA and Allocation are separately downloadable under their respective products. It is only the database schema component of these applications that is included with the RMS release.

Create Staging Directory for RMS Installer

To create the staging directory for RMS installer, complete the following steps.

Note: The same installer can be used to install multiple RMS components. If you are installing any of the RMS components (Database, Batch, or Application) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS installation software.
3. Copy the rms15installer.zip file from the RMS 15.0.2 release to the staging directory. This is referred to as STAGING_DIR when installing database software.

4. Change directories to STAGING_DIR and extract the rms15installer.zip file. This creates an rms/installer/ subdirectory under STAGING_DIR.

Note: The DB Schema and Batch install can be run at the same time, with the same installer, since they are configured to run from the database server. To run both, please follow instructions from the DB Schema Full install and Batch Full install sections of the install guide. This will ensure that both DB Schema and batch have the same RETAIL_HOME. When running the installer, select the Install Schema and Install batch check boxes.

Establish a Database Partitioning Strategy

Partitioning is mandatory for specific tables. Review this entire section before proceeding with the installation.

Note: Ensure the installer is used to automatically run the partition.ksh script when using the Sample Partitioning strategy. **Do not** run partition.ksh manually unless steps 1 and 2 below have been completed fully for the tables you wanted partitioned.

Sample Partitioning

The RMS 15.0.2 database schema installation runs the partitioning script (partition.ksh) automatically using a sample partitioning strategy if you do not run the partition script yourself. This is acceptable for development or demo installations and allows for a simpler installation. However, the resulting partitioning strategy is **not** suitable for production environments. It is highly recommended that the Production Partitioning section below be followed rather than allowing the installer to implement the sample strategy. The installer can be used to install the RMS database schema regardless of the choice made here.

Production Partitioning

Requirements for mandatory and optional partitioning are defined in the Microsoft Excel spreadsheet located in STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/RMS_partition_definition.xlsx. Since partitioning strategies are complex, this step should be implemented by an experienced individual who has a thorough understanding of partitioning principles and the data to be partitioned.

Use the Microsoft Excel spreadsheet to determine an appropriate partitioning strategy (STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/RMS_partition_definition.xlsx). The Partition Method column indicates the recommended partitioning options for each table. Refer to the information in this file to modify the DDL for partitioned tables. This can be done by manually changing the file STAGING_DIR/rms/installer/mom15/Cross_Pillar/ddl/1_rms_tab_ddl.sql or by implementing the process defined below.. This file will be used later in the installation process.

Note: Refer to Oracle Database Concepts 12 c Release 1 (12.1) Chapter 4 “Partitions, Views, and Other Schema Objects” for further details regarding partitioning concepts.

Beginning with hash partitions, complete the following process.

Hash partitions: To calculate the number of hash partitions and sub-partitions, enter values for the three parameters highlighted in yellow at the top of the RMS worksheet. Altering these values will update the Number of Partitions column for HASH partitioned/sub-partitioned tables. The values in these columns indicate the number of hash partitions/sub-partitions to create. Keep in mind that the number of hash partitions should be a power of 2.

Partition Factor: This value is used to adjust the number of hash partitions. It is based on the number of active items per location and transactions per location/day. If the number of items/location and/or transactions/store/day is low, the value of partition factor should be high. This will calculate fewer hash partitions. The typical factor value ranges from 2 to 4; in some cases, it can be 10 or more.

Note: Changing the items/location and transactions/store/day fields on the worksheet does not automatically impact the factor value. They are used as a point of reference only.

Sub-Partition Factor: This value is used to adjust the number of hash sub-partitions. The partition strategy for historical information determines the value of this number. If the number of range partitions is high, the value of sub-partition factor should be high to control the number of sub-partitions. Typically, this value is 2.

Locations: The total number of active stores and warehouses.

Range partitions: Determine the purging strategy for all of the tables that are RANGE partitioned. Each partition should have a range of multiple key values. For example, if the strategy were to have data available for one year and to purge it every three months, five partitions would be created. In this case, four 3-month partitions and a max value partition to contain all data greater than the defined ranges would result. Refer to the Comments column and update the value in the Number of Partitions column. The value in this column indicates the number of range partitions to create.

Interval partitions: Interval partitioning is an extension of range partitioning wherein the database automatically creates interval partitions as data for that partition is inserted. Determine the purging strategy for all of the tables that are INTERVAL partitioned. Each partition should have a range of multiple key values. For example, if the strategy were to have data available for 90 days and to purge it every week, you can create one 7 day partition, with an interval of 7 days. In this case, one 7 day partition would be created and any data that is inserted past the initial 7 day range will have a new partition automatically create to store the new data. Refer to the Comments column and update the value in the Number of Partitions column. The value in this column indicates the number of initial range partitions to create.

List partitions: The DAILY_ITEM_FORECAST, ITEM_FORECAST, DEAL_ITEMLOC_DCS, DEAL_ITEMLOC_DIV_GRP, DEAL_ITEMLOC_ITEM, AND DEAL_ITEMLOC_PARENT_DIFF must be LIST partitioned. If number of partition keys is relatively static, change the value in the Partition Method column to LIST where allowed. This method will ensure that each partition key has a separate partition and that none are empty. The Number of Partitions column will be automatically updated with the proper number of locations in the event the partition method is changed. The value in this column indicates the number of list partitions to create.

Step 1: Modify partition_attributes.cfg

Modify

STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/partition_attributes.cfg based on the partitioning strategy defined in RMS_partition_definition.xlsx. Changes to this file should be made only as indicated. partition_attributes.cfg file: (file is comma-delimited)

Sample Entry:

```
ITEM_LOC_HIST,EOW_DATE,RANGE,item_loc_hist.eow_date.date,64,LOC,HASH,item_loc_hist.loc.number,64,RETAIL_DATA
```

Field 1: Table Name - *Do not modify*

Field 2: Partition Key - *Do not modify*

Field 3: Partition Method - Modify based on value in Partition Method column in RMS_partition_definition.xlsx - Valid values are RANGE, LIST, HASH, or INTERVAL (case sensitive)

Field 4: Partition Data Definition Filename - *Do not modify - This field is ignored if Partition Method is not RANGE or LIST or INTERVAL*

Field 5: Partition Hash Count - Modify based on value in Hash Partitions Calculated column in RMS_partition_definition.xlsx. In case of INTERVAL partition, this field will contain a partition interval value (e.g. 7 days in one partition). *This field is ignored if Partition Method is not HASH or INTERVAL.*

Field 6: Interval Unit - Used and required for INTERVAL partition only. Expected values are 'DAY' or 'MONTH'.

Field 7: Sub-Partition Key - *Do not modify*

Field 8: Sub-Partition Method - Modify based on value in Sub-partition Method column in RMS_partition_definition.xlsx - Valid values are LIST or HASH (case sensitive)

Field 9: Sub-Partition Data Definition Filename - *Do not modify - This field is ignored if Sub-Partition Method is not RANGE, LIST, or INTERVAL*

Field 10: Sub-Partition Hash Count - Modify based on value in Hash Sub-partitions Calculated column in RMS_partition_definition.xls. *This field is ignored if Sub-Partition Method is HASH*

Field 11: Tablespace Name - *Optional. Default is RETAIL_DATA*

Step 2: Modify Data Definition Files

Tables partitioned or sub-partitioned by RANGE, INTERVAL or LIST have a corresponding data definition file in the STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/data_def directory and should not be removed or renamed. These files are used to define the data boundaries for each partition. Values must be entered in each file based on the data type of the Partition Key column in RMS_partition_definition.xls. Refer to the Comments column in this file for additional information. The value in the Number of Partitions column indicates the number of entries to place in the data definition file. For INTERVAL partitioning, a single entry in the data definition file will be sufficient.

The format of a data definition file name is <table name>.<partition key column>.<partition key data type> (for example, item_loc_hist.eow_date.date). When placing data into these files, enter one data partition value per line.

When entering varchar2 values in a data definition file, do not use quotation marks. When defining date values, use the DDMMYYYY format.

sampletable.action_date.date:

```
01012004
01012005
```

sampletable.state varchar2:

```
Minnesota
Iowa
```

sampletable.location.number:

```
1000
2000
```

When using RANGE partitioning, the data definition files will use the value less than concept. For example, in sampletable.action_date.date above, the first partition will contain all data less than 01012004. The second partition will contain all data greater than or equal to 01012004 and less than 01012005. A third MAXVALUE partition will automatically be created for all data greater than or equal to 01012005.

When using INTERVAL partitioning, the data definition file can be populated with one date entry to create the first range. Future partitions will be added automatically when data is inserted into the table for dates greater than the defined range and corresponding interval.

When using LIST partitioning, the data definition files will use the value equal to concept. For example, in sampletable.state varchar2 above, the first partition will contain all data equal to Minnesota. The second partition will contain all data equal to Iowa.

Step 3: Generate DDL for DAS Tables – Run partition.ksh (Optional)

1. Copy
STAGING_DIR/rms/installer/mom15/Cross_Pillar/das_ddl/source/rms_das_ddl.sql to
STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/rms_das.tab.
2. Execute
STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/partition_das.ksh at the UNIX command prompt. This script reads configuration information from the partition_attributes.cfg file and generates the partitioned DDL file STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/rms_das_part.tab. This file is used later during the installation process.

Sample output from partition.ksh:

```
STAGING_DIR/installer/mom15/Cross_Pillar/partitioning/source >
./partition.ksh
#####
###
# partition_das.ksh:
# This script will read the partition_attributes.cfg file and any
referenced
# data definition files and generate partitioned DDL.
#####
###
# The non-partitioned DDL file is ../rms_das.tab.
# The partitioned DDL file that will be generated is
../rms_das_part.tab.
```

```
#####
###
Checking partition_attributes.cfg for errors
Generating Partitioned DDL for ITEM_LOC
Generating Partitioned DDL for ITEM_LOC_SOH
Partition_das.ksh has generated the DDL for partitioned tables in the
../rms_das_part.tab file.
Completed successfully
```

3. Copy

STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/rms_das_part.tab to
STAGING_DIR/rms/installer/mom15/Cross_Pillar/das_ddl/source/rms_das_ddl.sql.

Step 4: Generate DDL for Tables – Run partition.ksh

1. Copy

STAGING_DIR/rms/installer/mom15/Cross_Pillar/ddl/1_rms_tab_ddl.sql to
STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/rms.tab.

2. Execute

STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/partition.ksh at the UNIX command prompt. This script reads configuration information from the partition_attributes.cfg file and generates the partitioned DDL file
STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/rms_part.tab.
This file is used later during the installation process.

Sample output from partition.ksh:

```
STAGING_DIR/installer/mom15/Cross_Pillar/partitioning/source >
./partition.ksh
#####
###
# partition.ksh:
# This script will read the partition_attributes.cfg file and any
referenced
# data definition files and generate partitioned DDL.
#####
###
# The non-partitioned DDL file is ../rms.tab.
# The partitioned DDL file that will be generated is ../rms_part.tab.
#####
###
Checking partition_attributes.cfg for errors
Generating Partitioned DDL for DAILY_DATA
Generating Partitioned DDL for DAILY_ITEM_FORECAST
Generating Partitioned DDL for DAILY_SALES_DISCOUNT
...
partition.ksh has generated the DDL for partitioned tables in the
../rms_part.tab file.
Completed successfully
```

Create the RMS Database

It is assumed that Oracle Enterprise Edition 12c Release 1, with appropriate patches, has already been installed. If not, refer to [Check Supported Database Server Requirements](#) in Chapter 1 before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in [Create Staging Directory for RMS Database Schema Files](#) in Part I, Chapter 1.

Review the [Establish Database Partitioning Strategy](#) section before continuing.

If a database has already been created, it is necessary to review the contents of this section to determine if all database components have been installed and configured properly. Also refer to appendices A, B, C in this document.

If a database instance has not been created, create one using database creation templates via DBCA in silent mode.

Create the Database Instance Using Oracle Generic Template

Prerequisites:

- 12.1.0.2 binary must have already been installed along with the appropriate oneoff patches. Refer to the Database Server Preinstallation section for all the required oneoff patchesBackground
- Oracle Retail no longer deliver custom database template files. Instead, databases can be created using the generic Oracle delivered template in the directory\$ORACLE_HOME/assistant/dbca/template.

▪

```
$ORACLE_HOME/assistant/dbca/templates>
--> ls -l General_Purpose.dbc
-rw-r--r-- 1 oracle rgbudba 4908 May 24 2013 General_Purpose.dbc
```

Instance Creation Using the Generic Template via DBCA

1. Ensure ORACLE_HOME and ORACLE_BASE is in the path:

```
export ORACLE_HOME=<Location for Oracle Home >
export ORACLE_BASE=<Location for Oracle Base>
export PATH=$ORACLE_HOME/bin:$PATH
.cd into $ORACLE_HOME/assistant/dbca/templates
```

2. Execute the following command to create an instance:

```
$ORACLE_HOME/bin/dbca -silent -createDatabase -templateName
General_Purpose.dbc -gdbName DB_NAME -sid DB_SID -
createAsContainerDatabase true -SysPassword oracle1 -
SystemPassword oracle1 -emConfiguration NONE -datafileDestination
<Datafile Location> -characterSet AL32UTF8 -nationalCharacterSet
AL16UTF16 -redoLogFileSize 100 -initParams nls_date_format=DD-MON-
RR,nls_language=AMERICAN,nls_calendar=GREGORIAN,fast_start_mttr_ta
rget=900
```

The above will create a container database using all the default parameters set by dbca. Please replace the pfile by taking a copy from [Appendix: Oracle 12cR1 Database Parameter File](#) but customize the values according to the need of your environment.

If you wish to create a non-container database, replace [-createAsContainerDatabase true] with [-createAsContainerDatabase false].

- Execute the following command to create a pluggable database if this is a container environment:

```
CREATE PLUGGABLE DATABASE PDB_NAME ADMIN USER PDBADMIN
IDENTIFIED BY pdbadmin_pwd ROLES=(CONNECT) file_name_convert=('<Old
Location of PDB Datafiles>', '<New Location for PDB Datafiles>');

alter pluggable database pdb_name open;

alter system register;
```

- Post Database Creation Setup

The above commands create a database with all files in one directory,. Please multiplex the redo logs and the control files following the OFA architecture.

- Configure the listener and the tnsnames entry.
- Log into the pluggable database to create the required tablespaces accordingly. For non-container databases, log into the database as normal to create the tablespaces.

Create Required RMS Tablespaces

Release 15.0 uses the tablespaces RETAIL_DATA, RETAIL_INDEX, ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX.

The ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX tablespaces hold data which may include Personally Identifiable Information data (PII Data). If you hold the Advanced Security Option license, you can choose to create these two tablespaces with TDE tablespace encryption to protect the PII data at rest. If you do not hold an Advanced Security Option license, you can create the tablespaces as normal tablespaces. The tablespace names must always be ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX regardless of whether TDE encryption is used, because the table and index creation scripts look for these specific names.

- Modify STAGING_DIR/rms/installer/create_db/create_rms_tablespaces.sql. The table below shows the default initial sizes.
- Once this script has been modified, execute it in SQL*Plus as sys.
 - For Example: SQL> @create_rms_tablespaces.sql
- Review create_rms_tablespaces.log for errors and correct as needed.
- If you do not wish to use TDE tablespace encryption follow below steps else for TDE encryption skip to step 5.
 - Modify STAGING_DIR/rms/installer/create_db/create_encrypted_tablespaces_no_TDE.sql.
 - Run the script using SQL*Plus as sys.
 - Review Create_encrypted_retail_tablespaces_no_TDE.log for errors and correct as needed.
- If you hold an Advanced Security Option license and wish to use TDE tablespace encryption
 - Modify STAGING_DIR/rms/installer/create_db/create_encrypted_tablespaces_TDE.sql.
 - Run the script using SQL*Plus as sys.

- c. Review `Create_encrypted_retail_tablespaces_TDE.log` for errors and correct as needed.
- d. Refer to [Appendix: Tablespace Creation](#) for details about how to create tablespaces in an encrypted format.

Note: The partitioning strategy determines the size of RMS tablespaces. Be aware that increasing the number of partitions may necessitate an increase in the size of the required tablespaces. It is important to be accurate when sizing tablespaces prior to the installation of RMS. Failure to do so results in “insufficient space” errors which require a complete re-install of RMS.

The standard tablespace scripts contain the DDL for creating the required tablespaces which can extend up to the following sizes:

TABLESPACE_NAME	Size
ENCRYPTED_RETAIL_INDEX	12G
ENCRYPTED_RETAIL_DATA	10G
RETAIL_INDEX	10G
RETAIL_DATA	8G
LOB_DATA	2G
USERS	2G

These sizes are sufficient if the initial values in the `STAGING_DIR/rms/installer/mom15/Cross_Pillar/partitioning/source/RMS_partition_definition.xls` spreadsheet are used without modifications. Although using the initial values is not recommended for a production environment, it is possible to use them for the purpose of creating a small test environment. For additional assistance with production database sizing, please work with your implementation partner or contact Oracle Retail Consulting.

Create the Schema Owner for RMS

Create an Oracle schema that will own the RMS application.

Note: The RMS schema owner must be created prior to running the RMS database schema installation. The installer will validate this user before proceeding with installation.

1. Change directories to `STAGING_DIR/rms/installer/create_db`.
2. The `create_user` script relies on empty roles, being created. Log into `sqlplus sys` as `sysdba` and run the following commands to create the roles.

```
SQL> @create_roles.sql
```

3. Enter the following command to create the schema owner:

```
SQL> @create_user.sql
```

The following prompts will occur:

- Schema Owner – the Oracle user that will own all RMS objects. Referred to in this install guide as RMS15DEV
 - Password – the password for RMS15DEV
 - Temp Tablespace – the temporary tablespace for RMS15DEV
4. Check the log file create_<Schema Owner>.lst for any errors.

Create the Database User RMS_ASYNC_USER

To create the required user RMS_ASYNC_USER, complete the following steps:

Note: RMS_ASYNC_USER is the required schema name, do not change the name to anything else. This user will own the RMS_NOTIFICATION_QUEUE used in the Async Notification

1. Change directories to STAGING_DIR/rms/installer/create_db
2. Log into sqlplus as sysdba and run the following command:
SQL> @create_rms_async_user.sql
The following prompts will occur:
 - Password – the password for RMS_ASYNC_USER
 - Temp Tablespace – the temporary tablespace for RMS_ASYNC_USER

Create the Database User for Allocation (Optional)

To create the database user for where Allocation temporary tables will be stored, complete the following steps.

1. Change directories to STAGING_DIR/rms/installer/create_db
2. Log into sqlplus as sysdba and run the following command:
SQL> @create_user_generic.sql
The following prompts will occur:
 - Schema Name – The name of the Allocation database user. Referred to in this install guide as ALLOC15DEV
 - Password – the password for ALLOC15DEV
 - Temp Tablespace – the temporary tablespace for ALLOC15DEV

Create the Database User for Demo Data (Optional)

The RMS demo data user is only required if you will be seeding RMS during installation with optional demo data. To create the demo data user, complete the following steps.

1. Change directories to STAGING_DIR/rms/installer/create_db
2. Log into sqlplus as sysdba and run the following command:
SQL> @create_user_generic.sql
The following prompts will occur:
 - Schema Name – The name of the Demo database user. Referred to in this install guide as RMS15DEMO
 - Password – the password for RMS15DEMO
 - Temp Tablespace – the temporary tablespace for RMS15DEMO

Create the Database User for DAS (Optional)

The RMS DAS user is only required if you will be setting up a DAS schema. Additional configuration of data replication will be required after installation. For an example of how to configure data replication using Oracle Streams, please refer to whitepaper: Oracle Retail White Paper: Configuring Data Replication for Retail Merchandising System Data Access Schema (DAS) using Oracle Streams

Note that the DAS user must be created in a different database from RMS. To create the DAS user, complete the following steps:

1. Change directories to STAGING_DIR/rms/installer/create_db
2. Log into sqlplus as sysdba and run the following command:

```
SQL> @create_user.sql
```

The following prompts will occur:

- Schema Name – The name of the DAS database user. Referred to in this install guide as RMS15DAS
- Password – the password for RMS15DAS
- Temp Tablespace – the temporary tablespace for RMS15DAS

Run the RMS Database Schema Installation

Note: See [Appendix: RMS Database Schema Installer Screens](#) for details on the RMS Database Schema installation screens and fields in the installer.

Note: It is recommended, but not required, that the Schema and Batch installation be done at the same time and use the same path for RETAIL_HOME. See next section for batch installation steps

1. Change directories to STAGING_DIR/rms/installer.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc).

```
Example: prompt$ . oraenv
           ORACLE_SID = [] ? mydb
           prompt$
```

3. Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

```
Example: prompt$ echo $ORACLE_HOME
           /u00/oracle/product/mydbversion
           prompt$ echo $ORACLE_SID
           mydb
```

4. Set and export the following environment variables. These variables are needed in addition to the environment variables set by the oraenv script above.

Variable	Description	Example
JAVA_HOME	Java home needed to run the GUI. Java 1.8 is required	JAVA_HOME=/usr/java/jdk1.8.64bit export JAVA_HOME
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
DISPLAY	Address and port of X server on desktop system of user running install. Optional for dbschema installation	DISPLAY=<IP address>:0.0 export DISPLAY

Note: Unset NLS_DATE_FORMAT before running the installer. If NLS_DATE_FORMAT is set as YYYY-MM-DD:HH24:MI:SS, the installer will fail.

5. If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
6. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh [text | silent]
```

7. Verify that the installer reports “SUCCESS” for the Database Preinstall Check. If it reports “FAILED,” check for errors in the output under the “Checking environment for Database installation” section, and verify that your environment variables are set properly.
8. For the initial RMS database installation select the Full option on the Full Install or Patch screen. If you are upgrading a previous install the Patch option will be used. See Part II: Upgrade Installation, Chapter 1: Database Installation Tasks - Upgrade.
9. Check the Install DB Objects checkbox and continue with installer. If the Batch and Database objects reside on the same RETAIL_HOME then click on the Batch also.
10. Depending on system resources, a typical installation can take two hours.
11. The RMS Installer provides the option of installing the Invoice Matching (ReIM) and Allocation database objects in addition to the RMS objects.
12. After the installer is complete, you can check its log file: rms-install.<timestamp>.log.
13. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example: `chmod 600 ant.install.properties`

Values to Remember for the Batch and Application Installation

If the RMS batch and application components will be installed separately, you will want to remember the database username and password details in order to correctly complete the RMS batch and application installations.

Resolving Errors Encountered During Database Schema Installation

If the installer encounters any errors, it halts execution immediately and prints to the screen which SQL script it was running when the error occurred. Please view the log files in \$RETAIL_HOME/orpatch/logs. Additional error information for invalid objects can be found in \$RETAIL_HOME/orpatch/logs/detail_logs/dbsql_{schema}/invalids. The {schema} refers to rms, rmsasync, reim, rpm, alloc, or alcrms.

See [Appendix: Common Installation Errors](#) in this document for a list of common installation errors.

Subsequent executions of the installer skip the SQL scripts which have already been executed in previous installer runs. This is possible because the installer maintains entries in a table called DBMANIFEST of the scripts that have been run. It also maintains an orpatch_restart.state file when the install restarts.

In case if you decided to drop the schemas and start the install from scratch, then make sure the RETAIL_HOME is also removed.

Set Up Additional RMS Users

1. Additional users to the RMS application can be set up by following the same instructions as in the Create the Schema Owner for RMS section above, except use the create_user_generic.sql script located here:
STAGING_DIR/rms/installer/create_db/create_user_generic.sql

Note: Evaluate the use of multiple roles and assign appropriately to users, based on user responsibilities.

2. After users are set up, create synonyms to the owner schema for all tables, views, sequences, functions, procedures, packages and types to which the user has access.

For information, see “[Appendix: Creating User Synonyms](#).”

3. Run the following scripts as the new user to give new users security privileges. These scripts can be found in the RMS installer package under STAGING_DIR/rms/installer/create_db. Sign on as the new user and run the following scripts.

```
SQL> @englishUser.sql
SQL> @superUser.sql
```

Note: create_ORMS_business_user_role.sql and create_ORMS_business_user.sql can be used to create RMS user with restricted privileges. Please refer to the *Oracle Retail Merchandising Operations Management Security Guide* for details.

Note: Users created with these scripts will be granted with selective privileges on each database object. A new object addition/patch that contains new objects will need attention from customer database administrator. Either grant selective privileges to the individual database objects or re-create the role with `create_ORMS_business_user_role.sql` which will grant privileges to new objects for the users.

PRODUCT_VERS_CONFIG_OPTIONS

1. Run the ad-hoc script as RMS Schema Owner `STAGING_DIR/rms/installer/mom15/Cross_Pillar/install_scripts/source/sys_update_prod_vers.sql` to update the `PRODUCT_VERS_CONFIG_OPTIONS` table. It updates the patch version of the other MOM products installed if any. It accepts five values as user input:
 - first input as Allocation version
 - second input as RWMS version
 - third input as REIM version
 - fourth input as SIM version
 - fifth input as AIP version
 - sixth input as RPM version
 - seventh input as ReSA version

Batch Security Setup

If RMS was installed without DEMO Data, additional data setup is required to be able to run batch programs. `USER_ATTRIB`, `SEC_USER`, `SEC_GROUP`, and `SEC_USER_GROUP` need to be populated using the below scripts.

1. Log on to sqlplus as the RMS schema owner.
2. Insert row into `SEC_GROUP`.


```
insert into sec_group (
GROUP_ID,
GROUP_NAME,
ROLE,
COMMENTS)
values (
1,
'SYSTEM SUPER USER GROUP',
NULL,
'Not to be associated with any hierarchy levels.');
```
3. Insert the following row into `USER_ATTRIB` for the schema owner:


```
@<STAGING_DIR>/rms/installer/create_db/englishUser.sql
```
4. Insert the following row into `SEC_USER` and `SEC_USER_GROUP` for the schema owner:


```
@<STAGING_DIR>/rms/installer/create_db/superUser.sql
```

Adding a User to the RPM Application

For LDAP authentication of RPM, complete the following steps to insert the user into the RMS schema.

1. Change directories to STAGING_DIR /rms/installer/mom15/rpm-db/install_scripts
2. Log on to sqlplus as the RMS schema owner.
3. Run the following script to insert row to database table:

```
@RSM_RPM_SE_user_role.sql <username>;
```

Example: @RSM_RPM_SE_user_role.sql RETAIL.USER;

Batch Installation Tasks—Full

This section includes steps for batch installation.

Create Staging Directory for RMS Installer

To create the staging directory for RMS installer, complete the following steps.

Note: The same installer can be used to install multiple RMS components. If you are installing any of the RMS components (Database, Batch, or Application) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS installation software.
3. Copy the rms15installer.zip file from the RMS 15.0.2 release to the staging directory. This is referred to as STAGING_DIR when installing batch software.
4. Change directories to STAGING_DIR and extract the rms15installer.zip file. This creates an rms/installer/ subdirectory under STAGING_DIR.

Note: Refer to the following My Oracle Support note if the operating system platform is Linux:

Doc ID 102288.1 – Precompiling Sample Pro*C Programs on Linux Fails with PCC-02015 and PCC-02201 (Doc ID 102288.1)

To fix the issue – Example:

1. Compare the paths in the installer pcscfg.cfg to the paths for pcscfg.cfg that the Linux OS has. The paths in the installer pcscfg.cfg are that may be invalid are
 - /usr/lib/gcc/x86_64-redhat-linux/4.1.2/include
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.6/include
 2. Find the pcscfg.cfg file in the correct path in the Linux OS. The path is
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.4
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.7 -> 4.4.4
 3. Back up the pcscfg.cfg file.
 4. Edit the pcscfg.cfg file.
 5. Change the following in the pcscfg.cfg file:
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.6/include to
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.7/include
 6. Run the batch installer.
-
-

Run the RMS Installer

To run the RMS Installer, complete the following steps:

Note: If Batch is installed along with Database installation then this step can be skipped.

Note: See [Appendix: RMS Batch Installation Screens](#) for details about the RMS Batch installation screens and fields in the installer.

1. Change directories to STAGING_DIR/rms/installer.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc).

Example: prompt\$. oraenv
ORACLE_SID = [] ? mydb
prompt\$

3. Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

Example: prompt\$ echo \$ORACLE_HOME
/u00/oracle/product/mydbversion
prompt\$ echo \$ORACLE_SID
mydb

4. Verify that the following executables are available from PATH: make, makedepend, cc, ar.

Example: Here are some locations where makedepend is commonly found:

Linux:	/usr/X11R6/bin
SUN:	/usr/openwin/bin
AIX:	/usr/X11R6/bin
HP-UX:	/opt/imate/bin

Set and export the following environment variables. These variables are needed in addition to the environment variables set by the oraenv script above.

Variable	Description	Example
JAVA_HOME	Java home needed to run the GUI. Java 1.8 is required	JAVA_HOME=/usr/java/jdk1.8.64 bit
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
DISPLAY	Address and port of X server on desktop system of user running install. Optional for batch installation	DISPLAY=<IP address>:0 export DISPLAY

5. If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
6. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh [text | silent]
```

7. Verify that the installer reports “SUCCESS” for the Batch preinstall check. If it reports “FAILED,” check for errors in the output under the “Checking environment for Batch installation” section, and verify that your environment variables are set properly.
8. Check the Install Batch checkbox and continue with installer.
9. Depending on system resources, a typical RMS batch installation takes around 30 minutes. After the installer is complete, you can check its log file in the “logs” directory: rms-install.<timestamp>.log.
10. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example: `chmod 600 ant.install.properties`

Resolving Errors Encountered During Batch Installation

The RMS batch installation is a full install that starts from the beginning each time it is run. If you encounter errors in your environment, after resolving the issue you can safely run the batch installation again to attempt another installation. Log files for the batch compilation can be found in the \$RETAIL_HOME/orpatch/logs/rmsbatch/{lib,proc}

Manual Steps for Running script ld_iindfiles.ksh

The prerequisite to using Item Induction is to load the templates on to the database tables. The templates drive the tables, columns to be loaded, and have the translation specific strings.

The below steps are required to load the templates into the environment. This is an optional step and is required only if the client chooses to implement Item Induction functionality.

1. Templates are present in
`<STAGING_DIR>/rms/installer/mom15/Cross_Pillar/s9t_templates` directory.
2. Review the template and include/exclude the details as required.
3. If not already set, export `TNS_ADMIN=<RETAIL_HOME>/orpatch/rms_wallet`
4. Go to `<RETAIL_HOME>/oracle/proc/src`.
5. Run `ld_iindfiles.ksh` by passing the two following parameters:
 - `UP=/@<schema owner wallet alias>`
 - Path to folder where the two ods files are located

```
ld_iindfiles.ksh $UP <STAGING_DIR>/rms/installer/mom15/Cross_Pillar/s9t_templates
```

RETL

The RMS batch installation installs the RETL files under RETAIL_HOME.

See [Appendix: RMS RETL Instructions](#) in this document for more information about RETL.

Data Conversion Scripts

The RMS batch installation installs the data conversion scripts under RETAIL_HOME/external/scripts. To complete the setup of these files, perform the following steps.

1. cd to RETAIL_HOME/external and chmod the permission on the following directories to 777:

```
RETAIL_HOME/external/data
RETAIL_HOME/external/logs
```

Note: You must replace RETAIL_HOME with your RETAIL_HOME directory.

2. Log into sqlplus as SYSTEM and run the following commands:

```
SQL> create or replace directory rms15dev_ext_data as
'RETAIL_HOME/external/data';
SQL> create or replace directory rms15dev_ext_logs as
'RETAIL_HOME/external/logs';
```

3. Log into sqlplus as SYSTEM and grant the following privileges to any other users who will be using data conversion.

```
SQL> grant read on directory rms15dev_ext_data to RMS15DEV;
SQL> grant read, write on directory rms15dev_ext_logs to RMS15DEV;
```

4. Update the following entries in the dc_load.cfg file in RETAIL_HOME/external/scripts:

```
export orclDataDir=RMS15DEV_EXT_DATA
export orclLogDir=RMS15DEV_EXT_LOGS
```

Application Server Installation Tasks – Full

It is assumed that WebLogic 12c version 12.2.1.2 (WLS) with Forms 12.2.1.2 has already been installed. If not, refer to “[Check Application Server Requirements](#)” in Chapter 1, “[Preinstallation Tasks](#)” before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in “[Create Staging Directory for RMS Application Files](#)” in Chapter 1.

In order to use WebLogic for manual compilation of RMS 15.0.2 forms modules, Oracle Forms Services 12c Release 2 (12.2.1.2) must be used. Please refer to the Oracle Forms Services 12c (12.2.1.2) documentation for the steps to manually compile objects.

Note: It is necessary to have \$ORACLE_HOME/network/admin/tnsnames.ora file configured in this WLS installation. Forms will use this information for connectivity.

A copy tnsnames.ora file must be created for the \$ORACLE_INSTANCE/config location. If the file is not copied to this location, forms will not compile correctly.

See [Appendix: Configure Listener for External Procedures](#) for an example of the tnsnames.ora file setup.

Prepare Application Server for RMS

Note: ORACLE_HOME is the location where Oracle Forms 12c has been installed.

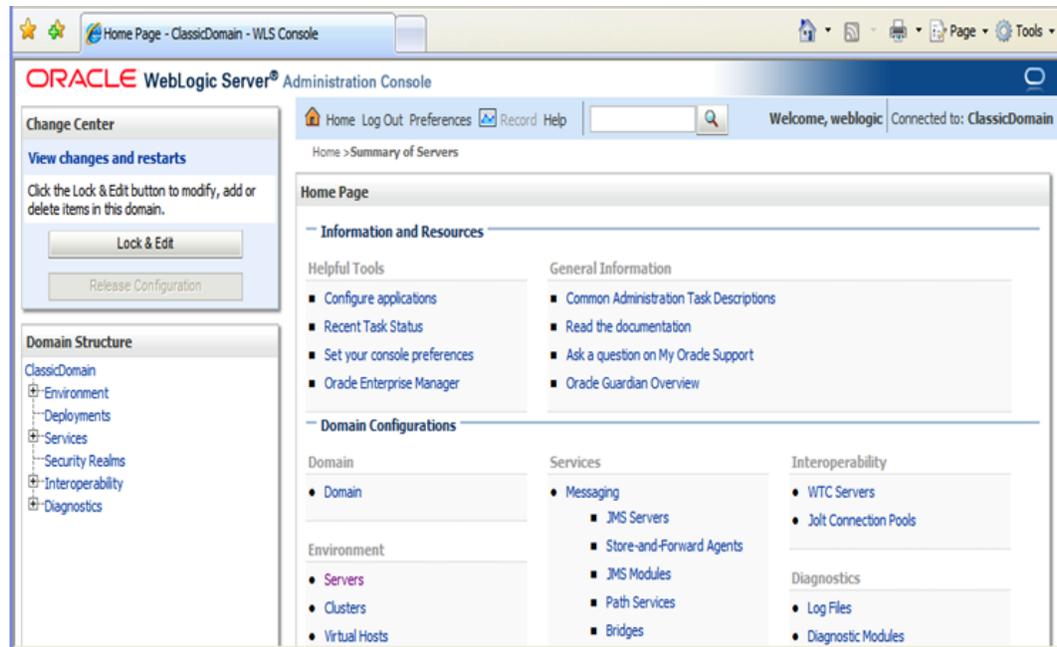
ORACLE_INSTANCE is the instance that is created during configuration of Oracle Forms 12c and contains the executables to compile.

To prepare the application server for RMS, complete the following steps.

1. The Tk2Motif.rgb file that is sent out with WebLogic must be modified. It is located at the following location: \$ORACLE_INSTANCE/guicommon/tk/admin
2. Make a copy of the file Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).
3. Modify the file Tk2Motif.rgb file so that it contains the following line:
`Tk2Motif*fontMapCs: iso8859-2=AL32UTF8`
4. Copy \$ORACLE_INSTANCE/guicommon/tk/admin/Tk2Motif.rgb to \$ORACLE_HOME/guicommon/tk/admin/Tk2Motif.rgb.

Create RMS Help Managed Server

1. Log in to the admin console.



2. Click **Lock & Edit**.

3. Navigate to Environment > Servers and select new tab of the servers on the right side.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Home > Summary of Servers

Create a New Server

Back Next Finish Cancel

Server Properties

The following properties will be used to identify your new server.

* Indicates required fields

What would you like to name your new server?

* **Server Name:** RMS-HELP

Where will this server listen for incoming connections?

Server Listen Address: apphost.us.oracle.com

* **Server Listen Port:** 7125

Should this server belong to a cluster?

No, this is a stand-alone server.

Yes, make this server a member of an existing cluster.

Select a cluster: cluster_forms

Yes, create a new cluster for this server.

Back Next Finish Cancel

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

ClassicDomain

- [-] Environment
- [-] Deployments
- [-] Services
- [-] Security Realms
- [-] Interoperability
- [-] Diagnostics

How do I...

- Create Managed Servers
- Configure default network connections

System Status

Health of Running Servers

Failed (0)

Critical (0)

4. Set the following variables:
 - **Server Name:** These should be some name specific to your application targeted (for example, rms-help-server).
 - **Server Listen Address:** <weblogic server> (ie apphost.us.oracle.com)
 - **Server Listen Port:** A free port; check for availability.

A suggestion is to increment the AdminServer port by two and keep incrementing by two for each managed server (for example, 7125, 7127, 7129, and so on).

5. Click Next.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Home > Summary of Servers

Create a New Server

Back Next Finish Cancel

Review choices

Review the selections. If these are correct, click Finish to create this server.

Server Name: RMS-HELP

Server Listen Address: apphost.us.oracle.com

Server Listen Port: 7125

Back Next Finish Cancel

6. Click Finish.

Click **Activate Changes** on the left side.

Install NodeManager

Install NodeManager if it was not created during domain install. NodeManager is required so that the managed servers can be started and stopped through the admin console. Only one node manager is needed per WebLogic installation.

1. Log in to the admin console.
2. Click **Lock & Edit** button and navigate to Environments > Machines.
3. Click **New**.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Home > Summary of Servers > Summary of Machines

Create a New Machine

Back Next Finish Cancel

Machine Identity

The following properties will be used to identify your new Machine.

* Indicates required fields

What would you like to name your new Machine?

* **Name:** apphost.us.oracle.com

Specify the type of machine operating system.

Machine OS: Unix

Back Next Finish Cancel

4. Set the following variables:
 - **Name:** Logical machine name
 - **Machine OS:** UNIX
5. Click **Next**.

6. Click on the machine created.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic | Connected to: ClassicDom

Messages

Machine created successfully

Summary of Machines

A machine is the logical representation of the computer that hosts one or more WebLogic Server instances (servers). WebLogic Server uses configured machine names to determine the optimum server in a cluster to which certain tasks, such as HTTP session replication, are delegated. The Administration Server uses the machine definition in conjunction with Node Manager to start remote servers.

This page displays key information about each machine that has been configured in the current WebLogic Server domain.

Customize this table

Name	Type
apphost.us.oracle.com	Unix Machine

Showing 1 to 2 of 2 Previous | Next

7. Click the Node Manager tab and update the details below.

- **Type:** Plain
- **Listen Address:** Host Name (apphost.us.oracle.com)
- **Listen Port:** Node manager will be assigned a default port (for example, 5556)

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Home > Summary of Servers > Summary of Machines > apphost.us.oracle.com

Settings for apphost.us.oracle.com

Configuration Monitoring Notes

General **Node Manager** Servers

Save

This page allows you to define the Node Manager configuration for this machine. To control a Managed Server from the console, the settings defined on this page are used to configure communication between the current domain and Node Manager instances.

Type: Plain

Listen Address: apphost.us.oracle.com

Listen Port: 5556

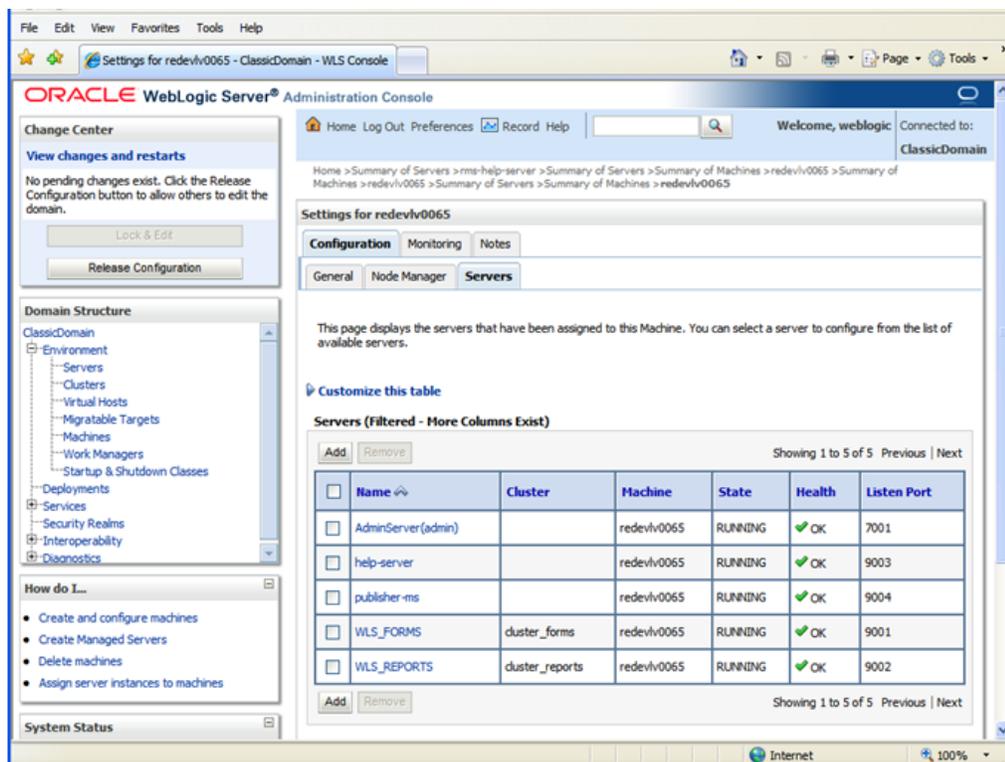
Node Manager Home:

Shell Command:

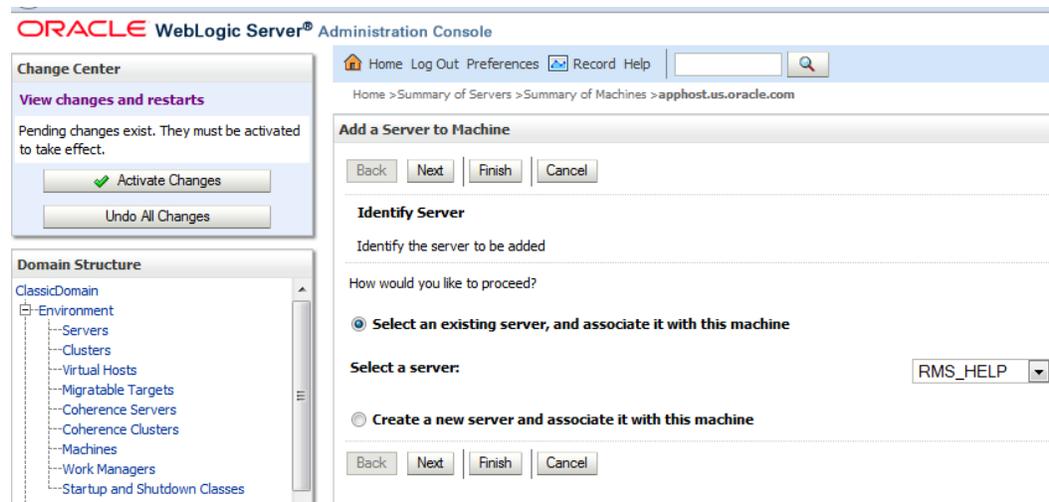
Debug Enabled

Save

8. Click **Save**.
9. Click **Activate Changes**.
10. Click **Lock & Edit**.
11. Navigate to **Environments->machines->click on the machine name and select the Servers tab**.



12. Click **Add**. Add the managed servers that need to be configured with NodeManager.



13. Set the following variables:
 - Server: name of server previously created (for example, rms-help-server)

14. Click **Next**. Click **Finish**.
15. Click **Activate Changes**.

Note: To activate changes the server must be stopped if it is running:

```
$WEBLOGIC_DOMAIN_HOME/bin/stopManagedWebLogic.sh <rms-help-server> ${server_name}:${server_port}
```

16. Start the Nodemanager from the server using the startNodeManager.sh at \$WEBLOGIC_DOMAIN_HOME/bin.

Create Staging Directory for RMS Application Server Files

To create the staging directory for the RMS Installer, complete the following steps.

Note: The same installer can be used to install multiple RMS components. If you are installing any of the RMS components (Database, Batch, or Application) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the application server as a user with read and write access to the WebLogic files.
2. Create a staging directory for the RMS installation software.
3. Copy the file rms15installer.zip from the RMS 15.0.2 release to staging directory. This will be referred to as STAGING_DIR when installing application software and reports.
4. Change directories to STAGING_DIR and extract the file rms15installer.zip. This will create an rms/installer subdirectory under STAGING_DIR.

Run the RMS Application Installation

Note: See [Appendix: RMS Application Installer Screens](#) for details about the RMS Forms application screens and fields in the installer.

1. Log on to your application server as a user with read and write access to the WebLogic files.
2. Change directories to STAGING_DIR/rms/installer.
3. Copy the following folders from ORACLE_HOME:
 - a. Copy jdk folder from ORACLE_HOME/oracle_common to ORACLE_HOME


```
cp -r /u00/webadmin/products/wls_forms/oracle_common/jdk
/u00/webadmin/products/wls_forms
```
 - b. Copy ojdbc7.jar from oracle_common/modules/oracle.jdbc/objdbc7.jar to the ORACLE_HOME/jdbc/lib folder


```
cp
/u00/webadmin/products/wls_forms/oracle_common/modules/oracle.jdbc/objdbc7.
jar /u00/webadmin/products/wls_forms/jdbc/lib
```

Note: ORACLE_HOME points to the forms installation home.

4. Set and export the following environment variables.

Variable	Description	Example
JAVA_HOME	Location of a Java 1.864Bit JDK.	JAVA_HOME= /u00/webadmin/java/jdk1.80_21 export JAVA_HOME
NLS_LANG	Locale setting for Oracle database client.	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
DOMAIN_HOME	The location where Forms 11.1.2.1 domain has been installed.	DOMAIN_HOME= /u00/webadmin/config/domains/wls_forms/ClassicDomain export DOMAIN_HOME
WLS_INSTANCE	The name of the managed server that contains Oracle Forms.	WLS_INSTANCE=WLS_FORMS export WLS_INSTANCE
ORACLE_HOME	Point to your WebLogic installation	ORACLE_HOME= /u00/webadmin/products/wls_forms export ORACLE_HOME
ORACLE_SID	The database/SID where the RMS schema resides.	ORACLE_SID=mydb export ORACLE_SID
DISPLAY	Address and port of X server on desktop system of user running install. Required for forms application installation.	DISPLAY=<IP address>:0 export DISPLAY

5. To install the RMS application you must use an X server (such as Exceed) and have set the DISPLAY environment variable. The installer does not continue otherwise.
6. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh [text | silent]
```

Verify that these environment variables are correct. If any of them are incorrect, you need to verify that the WebLogic shell scripts that set them are configured properly. Check the following scripts:

```
$WEBLOGIC_DOMAIN_HOME/bin/setDomainEnv.sh
$WEBLOGIC_HOME/oracle_common/common/bin/commEnv.sh
```

7. Verify that the installer reports “SUCCESS” for the Application preinstall check. If it reports “FAILED,” check for errors in the output under the “Checking environment for Application installation” section, and verify that your environment variables are set properly.
8. Check the Install Application checkbox and proceed with the installation.

Depending on system resources, a typical installation can take an hour or more.

Note: You may see the following warning repeated during installation:

```
[exec] Warning! One or more of your selected
locales are not available.
[exec] Please invoke the commands "locale" and
"locale -a" to verify your
[exec] selections and the available locales.
[exec]
[exec] Continuing processing using the "C" locale.
```

Or

```
[exec] couldn't set locale correctly
```

This warning can be ignored.

9. After the installer is complete, you can check its log file in the “logs” directory:


```
STAGING_DIR/rms/installer/logs/rms-install.<timestamp>.log.
RETAIL_HOME/orpatch/logs/detail_log/{forms/oraforms_rms}
```
10. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example: `chmod 600 ant.install.properties`

11. If during the screens you chose not to have the installer automatically configure WebLogic, after the installation is complete the files that would have been updated by the installer will be located in a directory and will have to manually be moved to correct directories as specified by the installer output like the example below.

Example:

```
#####
Weblogic Configuration Tasks
#####
Contact your Weblogic administrator and have them make backups of the following
files:
/u00/webadmin/config/domains/wls_forms/ClassicDomain
./config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config/oracle/fo
rms/registry/ Registry.dat
/u00/webadmin/config/domains/wls_forms/ClassicDomain
./config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config/formsweb.
cfg
```

Have the Weblogic administrator stop WLS_FORMS

```
copy everything in /u00/oretail/rms15/tst/post/MW_HOME
to /u00/webadmin/products/wls_forms
and copy everything in /u00/oretail/rms15/tst/post/DOMAIN_HOME
to /u00/webadmin/config/domains/wls_forms/ClassicDomain
to update the files
and then start WLS_FORMS
for the changes to take effect.
```

example:

```
cd /u00/oretail/rms15/tst/post/MW_HOME
cp -R * /u00/webadmin/products/wls_forms
cd /u00/oretail/rms15/tst/post/DOMAIN_HOME
cp -R * /u00/webadmin/config/domains/wls_forms/ClassicDomain
```

Verifying FORMS Configuration File Details

Once the installation is done, the formsweb.cfg located at <FORMS_DOMAIN_HOME>/config/fmwconfig/servers/<WLS_INSTANCE>/applications/formsapp_12.2.1/config should have the RMS configured as an example below:

```
[rms15]
envfile=./rms15inst/rms15inst.env
width=950
height=685
separateFrame=true
form=rtkstrt.fmx
archive=frmall.jar,rms-icons.jar,frmwebutil.jar,jacob.jar
lookAndFeel=Oracle
colorScheme=swan
```

If Single Sign-On is to be used with RMS, do the following.

- Set ssoMode to true.

If Resource Access Descriptors are allowed to be dynamically created,

Set ssoDynamicResourceCreate to true.

rms15inst.env mentioned above should have the following variables set:

```
NLS_DATE_FORMAT=DD-MON-RR
NLS_LANG=AMERICAN_AMERICA.AL32UTF8
FORMS_REJECT_GO_DISABLED_ITEM=FALSE
```

If In-Context Launch is not allowed, do the following.

- Set restrictedURLchars=userid

For more details, refer to the Application Specific Feature Administration section of the *Oracle Retail Merchandising Security Guide*.

Resolving Errors Encountered During Application Installation

In the event a form or menu does not compile, log files will be written to the \$RETAIL_HOME/orpatch/logs/detail_logs/forms directory. To try and manually recompile the object run RETAIL_HOME/base/forms.profile and run the following command:

```
# frmcmp.sh userid=$UP module_type=form module=FORM_OR_MENU
```

You can also safely rerun the installer to see if the form compiles.

Note: If you rerun the installer and check the **Configure WebLogic** box in the installer screens, you may need to clean up duplicate entries in the WebLogic formsweb.cfg file.

Webutil Setup

Webutil setup is required for doing item induction through the RMS application. The below setup will allow RMS to launching the file chooser to locate the spreadsheets to update to the application.

1. On the application server, change directories to \$ORACLE_HOME/forms.
2. Set and export the following environment variables.

Variable	Description	Example
ORACLE_HOME	Point to your WebLogic installation	ORACLE_HOME= /u00/webadmin/products/wls_forms export ORACLE_HOME
UP	Connect string of the RMS schema owner	UP= /@RMS01_mydb export UP
TNS_ADMIN	(Optional) If using wallet alias for connect string in "UP" this must be set the path to the directory containing the wallet	TNS_ADMIN=RETAIL_HOME/orpatch /rms_wallet_app/ export TNS_ADMIN
PATH	The location of sqlplus under ORACLE_HOME and frmcmp.sh under ORACLE_INSTANCE must be added to your PATH	PATH=/u00/webadmin/products/wls_forms/bin: /u00/webadmin/config/domains/wls_forms/FormsDomain/config/fmwconfig/components/FORMS/instances/form s1:\$PATH export PATH
DISPLAY	Address and port of X server on desktop system of user running install.	DISPLAY=<IP address>:0 export DISPLAY

3. Start a sqlplus session and run the sql script create_webutil_db.sql as the RMS schema owner.

```
sqlplus $UP
```

```
SQL> @create_webutil_db.sql
```

4. Compile webutil.pll and generate the plx.
`frmcmp.sh module=webutil.pll module_type=LIBRARY userid=$UP`
5. Verify the formsweb.cfg file includes jacob.jar and frmwebutil.jar on the archive line of RMS. This file is typically found within the WebLogic domain configuration folder in the subdirectory
.../fmwconfig/servers/<SERVER_NAME>/applications/formsapp/config

For example:

```
/u00/webadmin/config/domains/wls_forms/FormsDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config  
archive=frmall.jar,rms-icons.jar,frmwebutil.jar,jacob.jar
```

6. Copy a signed jacob.jar to \$ORACLE_HOME/forms/java.

Note: As a post install step to the webutil install - an additional update will need to be made to the jacob.jar due to enhanced java security requirements. The jacob.jar needs to be updated as follows:

- Unjar the jacob.jar
 - Remove all the signature information from the manifest
 - Add in the following lines of code to the manifest file:
 - Permissions: all-permissions
 - Codebase: *
 - Jar the contents back up again
 - Resign the jar.
-
-

Note: The jacob.jar file is not an Oracle supplied jar file, hence it requires signing with a trusted certificate from a Certificate Authority or a self-signed certificate to function properly with webutil. Failure to sign the jacob.jar file with an RSA certificate will result in the application not working. The following link gives instructions on how to properly sign the jacob.jar.
http://docs.oracle.com/javase/7/docs/technotes/guides/plugin/developer_guide/rsa_signing.html

7. Modify webutil.cfg file. This file is found in \$ORACLE_INSTANCE/config/FormsComponnet/forms/server. Find the line with transfer.database.enabled and change the value to TRUE.
`transfer.database.enabled=TRUE`

8. Bounce the admin server to enable these changes.

Clustered Installations – Post-Installation Steps

If you are installing the RMS application to a clustered environment, there are some extra steps you need to take to complete the installation. In these instructions, the application server node with the ORACLE_HOME you used for the RMS application installation is referred to as *master node*. All other nodes are referred to as *remote nodes*.

To complete the RMS forms application install, the installer provided new versions of formsweb.cfg and the newly-created env files for the new RMS installation. The entries added to formsweb.cfg and env files for these new environments should be copied from the master node to the remote nodes.

Note: The newly created env files will have a change to the FORMS_PATH variable as well as entries appended to the end of the file.

Note: Do not copy the entire file from one node to another. Only copy the RMS entries modified in these files by the installer. There is node-specific information in this file that is different between ORACLE_HOME installations..

RMS Reports Copied by the Application Installation

The application installation copies RMS report files to \$RETAIL_HOME/base/reports. These files should be installed into BI Publisher as documented in the RMS Reports chapter of this document.

Oracle BI EE Configuration for RMS Reports

RMS 15.0.2 reports supports BiPublisher 12c. RMS Reports are copied to RETAIL_HOME /reports during the application installation.

Note: In the following sections, the Oracle BI EE 12c installation steps are a sample only. Refer to the *Oracle Business Intelligence 12c Installation Guide* for more information.

BI Server Component Installation Tasks

Oracle BI Publisher is used as the main RMS, RWMS, REIM, and SIM reporting engine and can be used in conjunction with external printing solutions like label printing. This section describes the installation of Oracle BI Publisher as a server application within WebLogic 12.2.1.2. One deployment of BI Publisher can be used for any of the RMS, RWMS, REIM, and SIM reports.

When installing BI Publisher 12g, refer to the appropriate Fusion Middleware guides for the installation of the product in a WebLogic server environment.

Installation Process Overview

Installing the BI Publisher server as a standalone web application in a WebLogic server involves the following tasks:

1. Run RCU to create BIPublisher related database schemas and other db objects.
2. Install Oracle BI EE using the "Enterprise Install" option.
3. Configure Oracle BI EE, create default bifoundation_domain and configure component "Business Intelligence Publisher" only.
4. Select the BIPlatform schema for update of the ORACLE 12c DB
5. Configure ports and document and test the URL's that are created.

The following post-installation tasks are involved once BI Publisher has been installed:

6. Set up and copy the RMS BI Publisher Report Templates produced for RMS.
7. Configure the BI Publisher repository. Set security model, add users, assign roles, add reports, add printers, set repository path, set data source, etc.
8. Set up for the RMS application specific configuration files to integrate BI Publisher with the RMS online app.

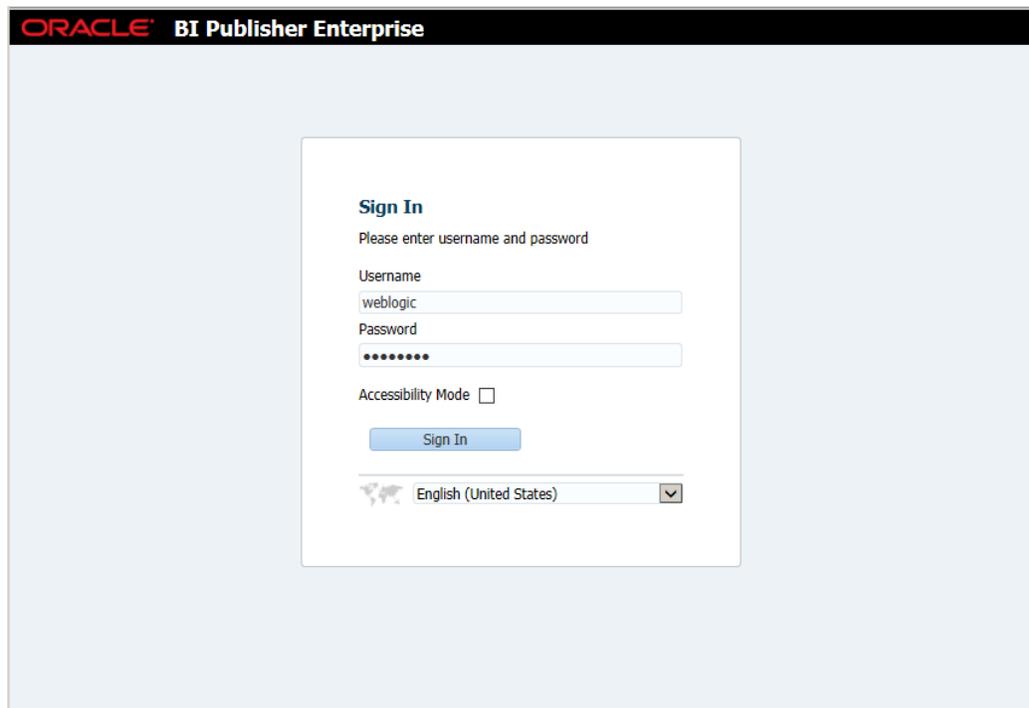
Post Install Steps for OBIEE 12C

Oracle Business Intelligence 12.2.1.2 is a unique platform that enables customers to uncover new insights and make faster, more informed business decisions by offering agile visual analytics and self-service discovery together with best-in-class enterprise analytics. Install and configure OBIEE 12C by using following link.

<http://docs.oracle.com/middleware/12212/core/BIEIG/GUID-04F89ACA-A2F1-4F18-8B35-BD131ACC62ED.htm#INSOA369>

Once installation is done by setting correct ORACLE_HOME and DOMAIN_HOME and the xmlpserver and analytics url's must be working properly before we start following post install steps.

1. Test your BIPublisher installation, Get the xmlpserver url from your Installation Screen and launch xmlpserver. Login with the credentials you entered in your Oracle BI EE configuration (weblogic / password). Example URL:[http://\[obiee_host\]:\[obiee_server_port\]/xmlpserver](http://[obiee_host]:[obiee_server_port]/xmlpserver)



ORACLE BI Publisher Enterprise

Sign In
Please enter username and password

Username
weblogic

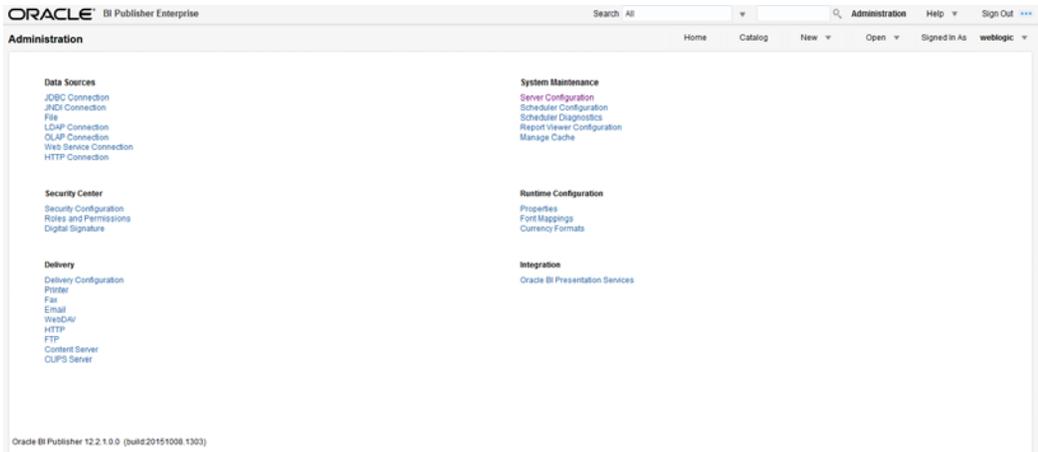
Password
••••••••

Accessibility Mode

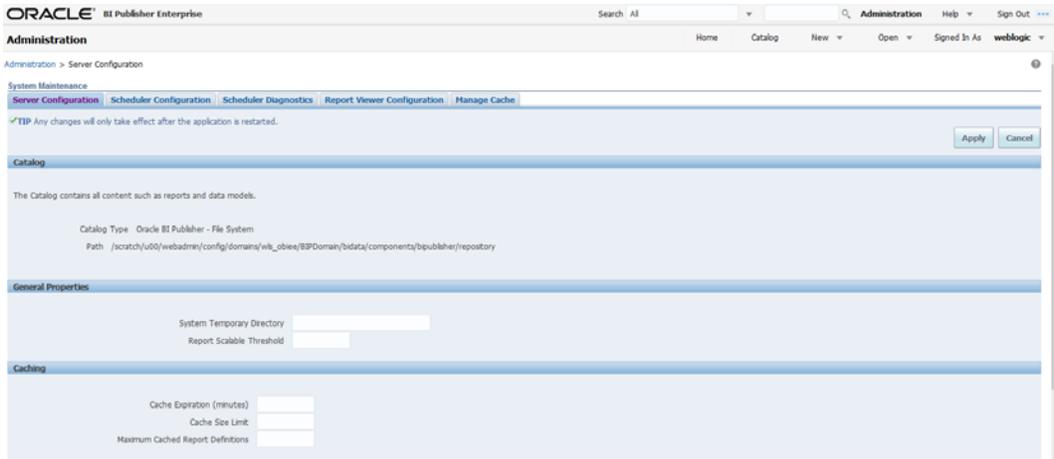
Sign In

English (United States)

2. After sign on, select "Administration".



3. On the System Maintenance Section, click **Server Configuration**.

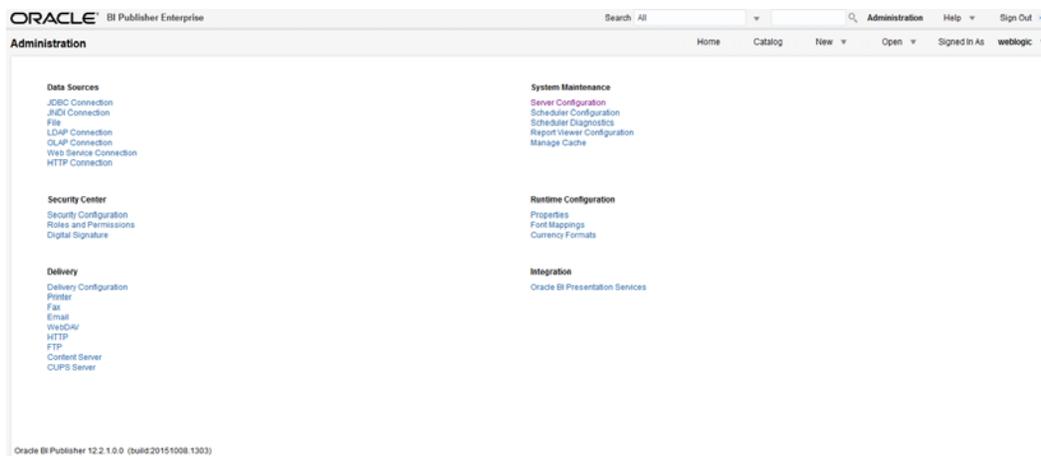


4. On this screen - In the Server Configuration Folder section, enter the path to your repository.

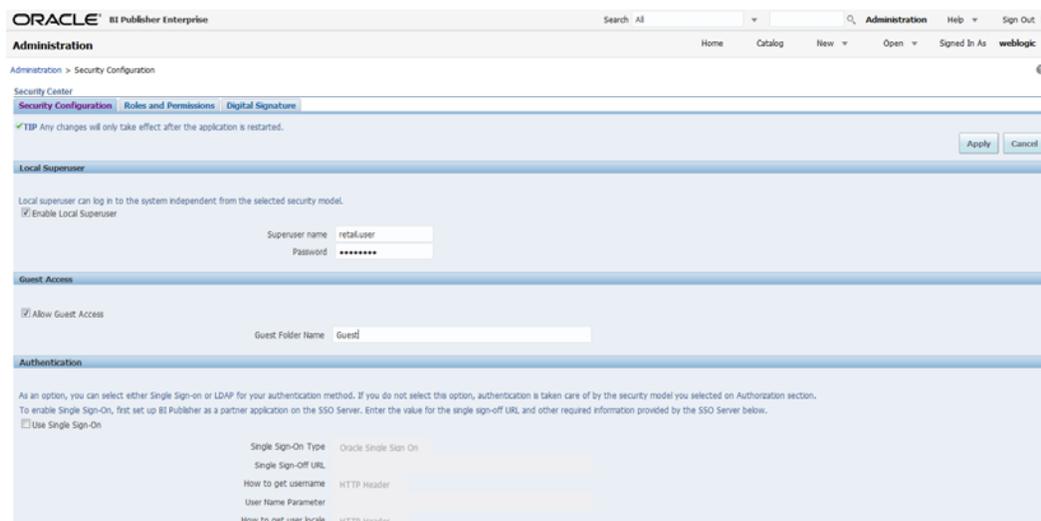
- This is the path you entered in the Configuration Section and Catalog Section:
Example: `$<OBIEE_DOMAIN_HOME>/bidata/components/bipublisher/repository`

5. Click **Apply**.

6. Click Administration link at top of screen.



7. Click the Security Configuration link under the Security Center to setup a super user and apply the BI Publisher security model.



8. Enable a Supersuser by checking the “Enable Local Supersuser” box and by entering name and password on the corresponding fields on this screen.
9. Mark “Allow Guest Access” check box. Enter “Guest” as Guest Folder Name.
10. Click **Apply**.
11. Scroll down the screen and locate the Authorization section:

ORACLE BI Publisher Enterprise

Administration

Home Catalog New Open Signed In As retailuser

Allow Guest Access

GuardFolder Name Guard

Authentication

As an option, you can select either single sign-on or LDAP for your authentication method. If you do not select this option, authentication is taken care of by the security model you selected in the Authentication section. To enable Single Sign-On, first set up BI Publisher as a partner application on the SSO Service. Enter the value for the single sign-on URL, and other required information provided by the SSO Service below.

Use Single Sign-On

Single Sign-On Type Oracle Single Sign-On

Single Sign-On URL

How to get username HTTP Header

User Name Parameter

How to get user locale HTTP Header

User Locale Parameter

Enter the value for URL, Administrator Username, Administrator Password, Distinguished Name for Users and other required information below.

Use LDAP

URL

Administrator Username

Administrator Password

Distinguished Name for Users

JNDI Context Factory Class

Attribute used for Login Username

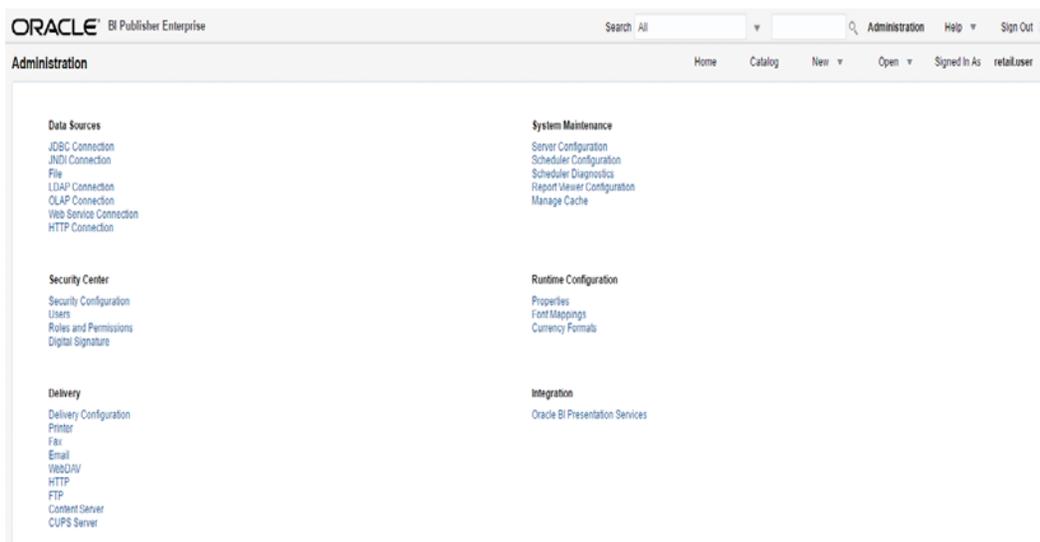
Attribute used for user matching with authorization system

Authentication

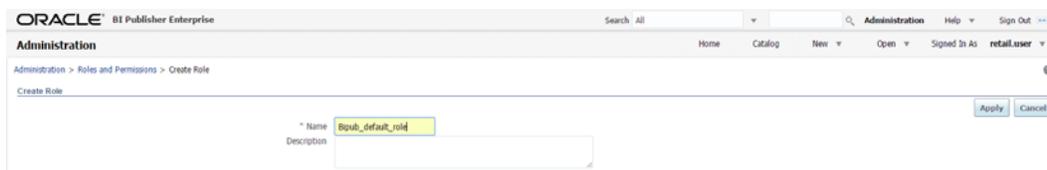
Security Model BI Publisher Security

New Password

12. Select BI Publisher Security from the Security Model list.
13. The default user name for the BI Publisher Security Model is Administrator in `xmpl-server-config.xml` file. The Local Super User enable you to bypass all the security configuration and to be able to manage BIP.
14. On the password text field, enter a value that you can remember. It is going to be the password for Login to `xmplserver`.
15. Click **Apply**.
 - Leave BI Publisher up while completing the next section.
16. Post install step: Create role `Bipub_default_role`.
 - a. From the `xmplserver` Administration screen, scroll down to Security Center and click Roles and Permissions.



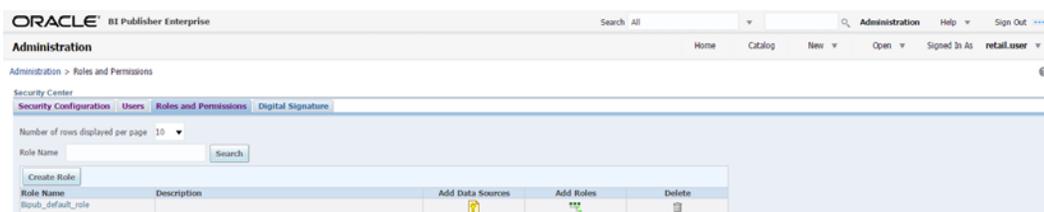
- b. On the Roles and Permissions screen, click the Create Role button.



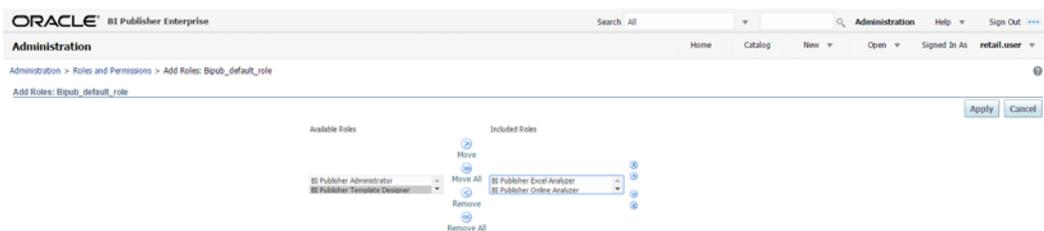
- c. Create the Bipub_default_role. Enter in Create Role Section name of the role.
- d. When the information has been entered press Apply changes.

17. Post install step:

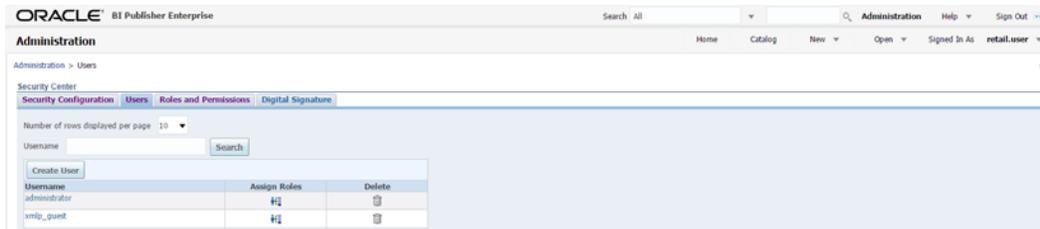
- a. Assign BiPublisher roles to the newly created Bipub_default_role. The BiPublisher roles used to control access to reports and data sources.
- b. To assign BiPub system roles to the newly create Bipub_default_role, go to Security Center section and navigate to the Roles and Permissions screen:



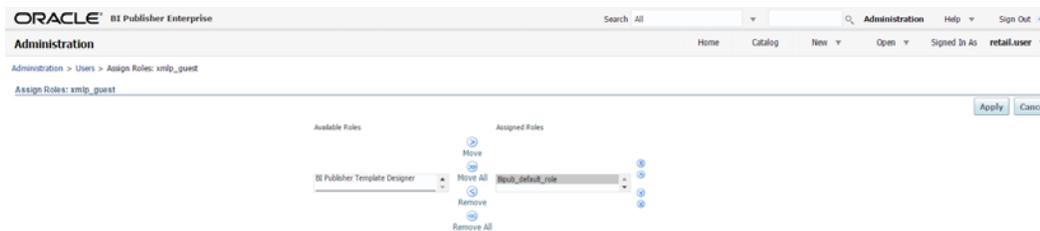
- c. On the Roles and Permissions screen you should see the new role created: "Bipub_default_role". Add multiple roles to the Bipub_Default_Role by pressing the corresponding green icon on the Add Roles column.
- d. BI Publisher Excel Analyzer, BI Publisher Online Analyzer, BI Publisher Scheduler.



- e. From the "Available Roles" panel, select the ones needed for your reports and move them to the "Included Roles" panel.
 - f. Press the Apply button to save your changes.
18. Post install step: create Guest (XMLP_GUEST) user.
- a. From the xmlpserver Administration screen scroll down to Security Center section and press Users to navigate to the next screen.



- b. Select the "Create User" button to create the "xmlp_guest" user and save the changes.
19. Post install step: Adding the Bipub_default_role to XMLP_GUEST user.
- a. Open the Users section:
 - b. For xmlp_guest user, press on the "Assign Roles" icon to navigate to the next screen:



- c. On the Assign Roles screen, select the BiPub_default_role from the Available Roles panel to the "Assigned Roles" panel and press the Apply button to save your changes.

Installing the RMS BI Publisher Templates

In this section we will outline how the RMS report templates are installed into the appropriate BI server repositories

Example: `$<OBIEE_DOMAIN_HOME>/bidata/components/bipublisher/repository`

The Installer copies the report templates in the directory - "RETAIL_HOME /reports" and have to be copied into a newly created directory within BI Publisher repository Guest Reports directory.

1. Create the directory to hold the reports under `<BI_REPOSITORY>`
`mkdir <BI_REPOSITORY>/Reports/Guest/RMS`
2. RETAIL_HOME is the location where reports get copied and defined before the installation begin. Change directory to the RETAIL_HOME /reports/RMS created during the application install. This directory contains subdirectories whose names reflect the names of report templates provided with RMS.
3. Copy each report directory into the directory created above
4. Go to RETAIL_HOME /reports/RMS and copy the reports to the below location.

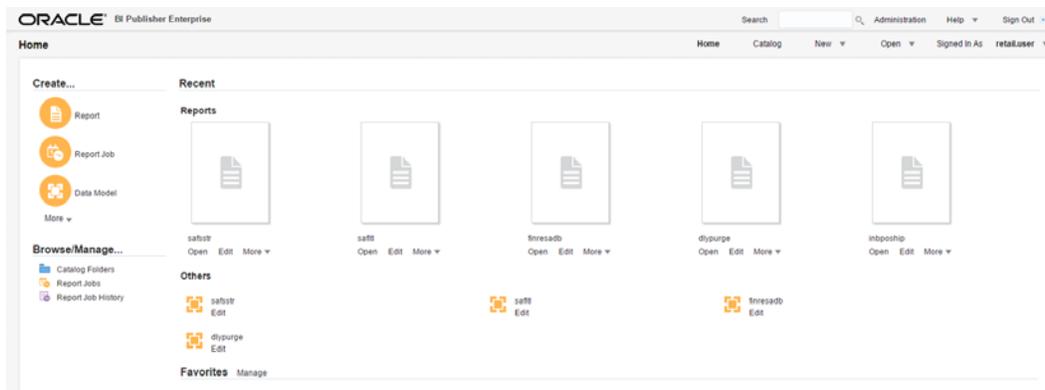
`cp -R * <BI_REPOSITORY>/Reports/Guest/RMS`

Configuring the RMS JDBC Connection

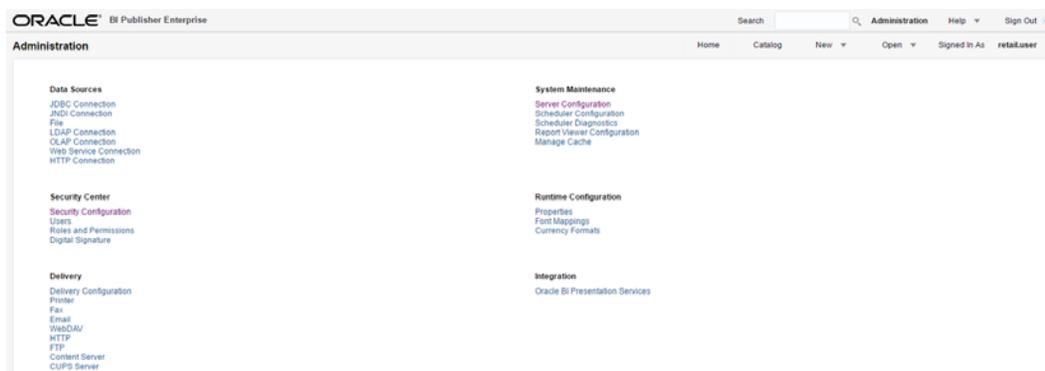
Follow the below steps to configure a JDBC connection for the RMS Data Source, which is required for RMS reports. Please check if needed add this Datasource.

1. Login as the super user that was created in prior security setup steps.

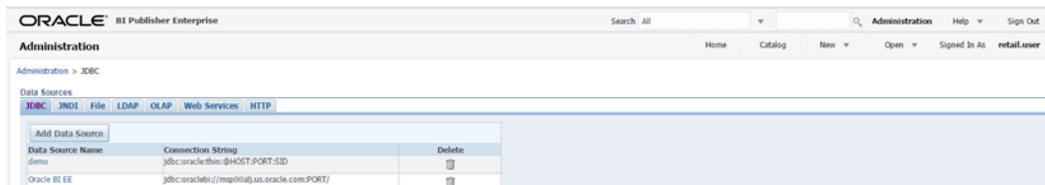
Note: You will not be able to login to xmlpserver as weblogic anymore because we have already changed the Security Model.



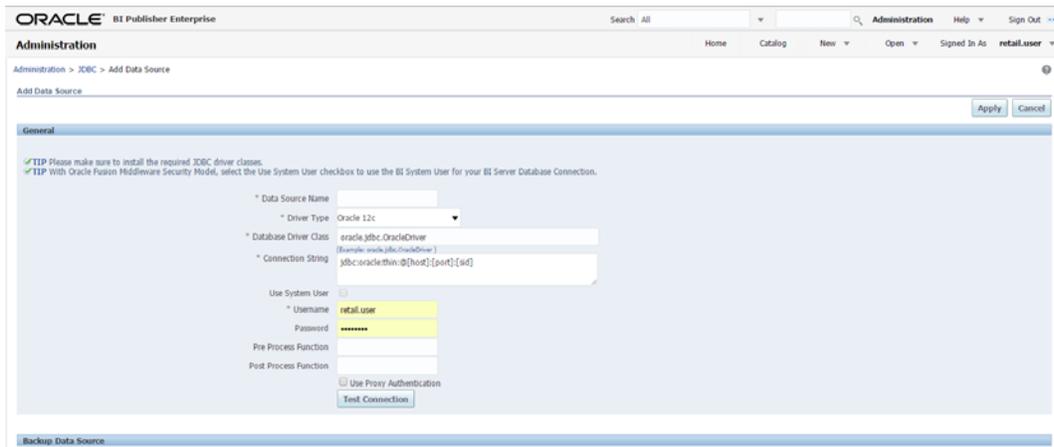
2. Click the **Administration** link at top of screen



3. Select the JDBC Connection hyperlink in the Data Sources lists.



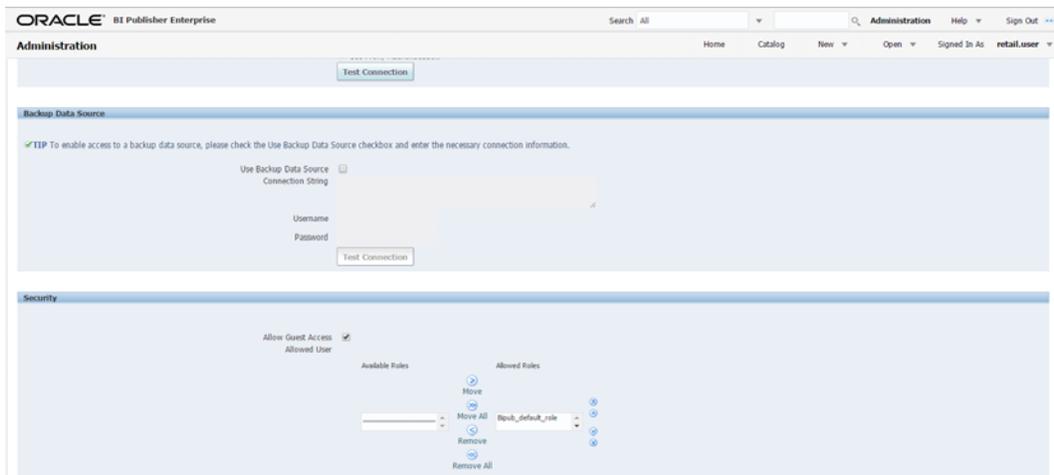
4. Click the **Add Data Source** button.



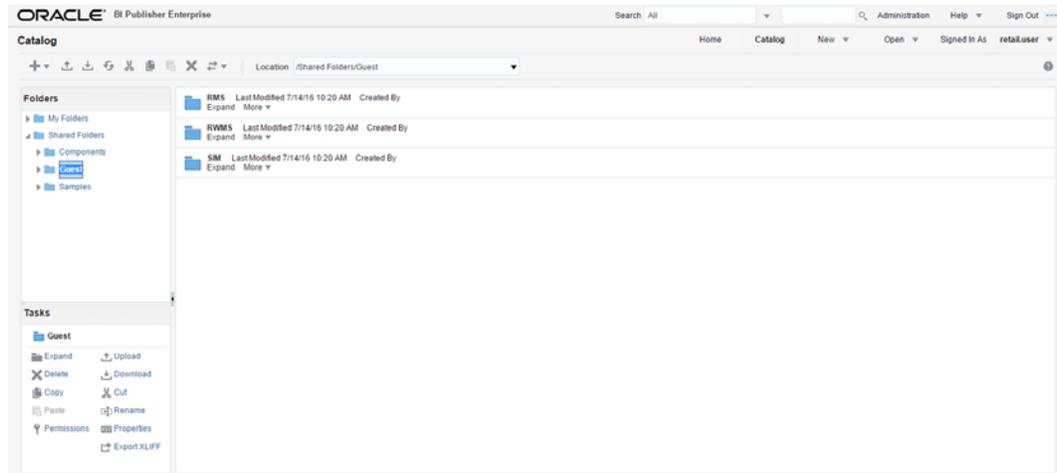
5. Enter the appropriate details for the RMS data source. Click **Test Connection** to test the connection on the screen once the data is entered.

- Data Source Name: RMS
 - Must be RMS due to code dependencies.
- Driver type is ORACLE 12C
- Database driver class should be oracle.jdbc.OracleDriver.
- Connection string is similar to this example:
 - Pluggable: jdbc:oracle:thin:@dbhostname:1521/servicename
 - Non- Pluggable dbc:oracle:thin:@dbhostname:1521:SID
- Enter the username and password for the RMS application user’s data source. Click **Test Connection** to test the connection on the screen once the data is entered.

6. Scroll to the bottom of the screen and check the **Allow Guest Access** check box. Click **Apply**.



7. Click Catalog link at the top of the screen – and then click the Guest folder on the left so that it is highlighted.



8. Click OK.
9. Restart WebLogic Server and verify reports.

Part II: Upgrade Installation

The database portion of RMS can be upgraded from a 15.0 release to release 15.0.2 Part II of this guide details the steps needed to perform an upgrade installation of RMS.

The *Oracle Retail Merchandising Upgrade Guide* describes the approach that this Oracle Retail application takes for the upgrading process, as well as this product's upgrade assumptions and considerations.

Part II contains the following chapters:

- [RMS Database Installation—Upgrade](#)
- [Batch Installation Tasks—Upgrade](#)
- [Application Server Installation Tasks—Upgrade](#)
- [Reports Installation Tasks—Upgrade](#)
- [Data Migration](#)

For information about a full installation, see Part I.

Database Installation Tasks – Upgrade

Upgrade RMS Database using the Installer

The RMS 15.0.2 database schema installer may be used to apply the RMS upgrade. The installer should only be used to apply the upgrade if the schema being upgraded does not contain customizations. In this section, STAGING_DIR refers to the location where the RMS 15.0.2 installer is expanded.

Before you apply the RMS 15.0.2 upgrade:

- Make a backup of all your objects and database schema.
- Check that RMS is installed and is at 15.0.1 level.
- Review each of the enclosed defect documents.
- Make sure any applications that connect to the RMS schema are shut down. This includes RPM, ReIM, Allocation, RIB, and anything else that could be using the schema.

The following are the staging tables which RPM owns that add/remove data during upgrade process. These tables need to be emptied before starting an upgrade.”

- RPM_STAGE_SIMPLE_PROMO
- RPM_STAGE_PRICE_CHANGE
- RPM_STAGE_CLEARANCE
- RPM_STAGE_CLEARANCE_RESET
- RPM_STAGE_THRESHOLD_PROMO
- RPM_STAGE_COMP_THRESH_LINK
- RPM_STAGE_MULTIBUY_BUYLIST
- RPM_STAGE_MULTIBUY_HEADER
- RPM_STAGE_MULTIBUY_RWDLIST
- RPM_STAGE_TRAN_PROMO_BUYLIST
- RPM_STAGE_TRAN_PROMO_HEADER
- RPM_STAGE_TRAN_PROMO_RWDLIST
- RPM_STAGE_FINANCE_PROMO
- RPM_STAGE_FIN_CRED_DTL
- RPM_STAGE_FIN_THRESH_DTL
- RPM_CLEARANCE_PAYLOAD
- RPM_FIN_CRED_DTL_PAYLOAD
- RPM_PRICE_CHG_PAYLOAD
- RPM_PRICE_EVENT_PAYLOAD
- RPM_PROMO_DISC_LDR_PAYLOAD
- RPM_PROMO_DTL_CIL_ITEM_PAYLOAD
- RPM_PROMO_DTL_CIL_LOC_PAYLOAD
- RPM_PROMO_DTL_CIL_PAYLOAD
- RPM_PROMO_DTL_LIST_GRP_PAYLOAD
- RPM_PROMO_DTL_LIST_PAYLOAD

- RPM_PROMO_DTL_MN_PAYLOAD
- RPM_PROMO_DTL_PAYLOAD
- RPM_PROMO_DTL_PRC_RNG_PAYLOAD
- RPM_PROMO_FIN_DTL_PAYLOAD
- RPM_PROMO_ITEM_LOC_SR_PAYLOAD
- RPM_PROMO_ITEM_PAYLOAD
- RPM_PROMO_LOCATION_PAYLOAD
- RPM_THRESHOLD_INT_PAYLOAD
- RPM_CC_SYS_GEN_DETAIL_WS
- RPM_CC_SYS_GEN_HEAD_WS
- RPM_CLEARANCE_WS
- RPM_CUST_SEGMENT_PROMO_FR_WS
- RPM_FUTURE_RETAIL_WS
- RPM_PROMO_ITEM_LOC_EXPL_WS
- RPM_BULK_CC_PE
- RPM_BULK_CC_PE_CHUNK
- RPM_BULK_CC_PE_ITEM
- RPM_BULK_CC_PE_ITEM_GTT
- RPM_BULK_CC_PE_LOCATION
- RPM_BULK_CC_PE_SEQUENCE
- RPM_BULK_CC_PE_THREAD
- RPM_BULK_CC_TASK

Create Staging Directory for RMS Database Schema Files

To create a staging directory for RMS database schema files, complete the following steps.

Note: The same installer can be used to upgrade multiple RMS components. If you are installing any of the RMS components (Database, Batch, or Application) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the MOM 15.0.2 Upgrade.
3. Copy the rms15installer.zip file from the RMS 15.0.2 release to the staging directory. This is referred to as STAGING_DIR when installing database software.
4. Change directories to STAGING_DIR and extract the rms15installer.zip file. This creates an rms/installer subdirectory under STAGING_DIR.

Optional- Analyze Changes in the Patch

Note: See [Appendix: RMS Analyze Tool](#) for details and instructions to run the RMS Analyze Tool. This appendix also contains screens and fields in the tool.

Run the Allocation Data Cleanup Scripts

If upgrading Allocation to 15.0.2, the following steps need to be completed to cleanup data before running the installer.

1. Change directories to STAGING_DIR/rms/installer/upgrade_scripts/alloc.
2. Log on to sqlplus as the RMS schema owner.
3. Run the following scripts:

```
@raf_cleanup.sql;
@rtc_lookup_cleanup_old_rows.sql;
```

Run the RMS Database Schema Upgrade

To run the RMS database schema upgrade, complete the following steps.

Note: See [Appendix: RMS Database Installation Screens](#) for details on the RMS Database Schema installation screens and fields in the installer.

For upgrade, ensure the schema names are entered same as that in the previous installations since wallet alias is case sensitive.

For clarification, refer
\$RETAIL_HOME/orpatch/rms_wallet path.

Example: <rms01_mydb>

1. Change directories to STAGING_DIR/rms/installer.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc)

Example: prompt\$. oraenv
ORACLE_SID = [] ? mydb

Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

Example: prompt\$ echo \$ORACLE_HOME
/u00/oracle/product/mydbversion
prompt\$ echo \$ORACLE_SID
mydb

3. Set and export the following environment variables. These variables are needed in addition to the environment variables set by the oraenv script above.

Variable	Description	Example
JAVA_HOME	Java home needed to run the GUI. Java 1.8 is required	JAVA_HOME=/usr/java/jdk1.8.64bit
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG

Variable	Description	Example
DISPLAY	Address and port of X server on desktop system of user running install. Optional for dbschema installation	DISPLAY=<IP address>:0 export DISPLAY

Note: Unset NLS_DATE_FORMAT before running the installer. If NLS_DATE_FORMAT is set as YYYY-MM-DD:HH24:MI:SS, the installer will fail.

- If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
- Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
install.sh [text | silent]
```

- Verify that the installer reports “SUCCESS” for the Database preinstall check. If it reports “FAILED,” check for errors in the output under the “Checking environment for Database installation” section, and verify that your environment variables are set properly
- Select the Patch option on the Full Install or Patch Option screen.
- Check the Install DB Objects checkbox and continue with installer. If the Batch and Database objects reside on the same RETAIL_HOME then click on the Batch also.
- On the RETAIL_HOME screen, select the RETAIL_HOME of your previous installation.
- On the Wallet password screen, enter the wallet password you used in the previous installation.
- After the installer is complete, you can check its log file: rms-install-dbschema.<timestamp>.log.
- The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example: `chmod 600 ant.install.properties`

Resolving Errors Encountered During Database Schema Installation

If the installer encounters any errors, it halts execution immediately and prints to the screen which SQL script it was running when the error occurred. Please view the log files in RETAIL_HOME/orpatch/logs. Additional error information for invalid objects can be found in RETAIL_HOME/orpatch/logs/detail_logs/dbsql_{schema}/invalids. The {schema} refers to rms, rmsasync, reim, rpm, alloc, or alcrms.

See [Appendix: Common Installation Errors](#) in this document for a list of common installation errors.

Subsequent executions of the installer skip the SQL scripts which have already been executed in previous installer runs. This is possible because the installer maintains entries in a table called DBMANIFEST of the scripts that have been run. It also maintains an orpatch_restart.state file when the install restarts.

In case if you decided to drop the schemas and start the install from scratch, then make sure the RETAIL_HOME is also removed.

Batch Installation Tasks—Upgrade

The RMS 15.0.2 installer may be used to upgrade the RMS batch. Before you apply the RMS 15.0.2 batch upgrade:

- Review the enclosed RMS 15.0.2 Upgrade Release Notes.

Create Staging Directory for RMS Installer

To create the staging directory for RMS installer, complete the following steps.

Note: The same installer can be used to install multiple RMS components. If you are installing any of the RMS components (Database, Batch, or Application) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS installation software.
3. Copy the rms15installer.zip file from the RMS 15.0.2 release to the staging directory. This is referred to as STAGING_DIR when installing batch software.
4. Change directories to STAGING_DIR and extract the rms15installer.zip file. This creates an rms/installer/ subdirectory under STAGING_DIR.

Note: Refer to the following My Oracle Support note if the operating system platform is Linux:

Doc ID 102288.1 – Precompiling Sample Pro*C Programs on Linux Fails with PCC-02015 and PCC-02201 (Doc ID 102288.1)

To fix the issue – Example:

1. Compare the paths in the installer pcscfg.cfg to the paths for pcscfg.cfg that the Linux OS has. The paths in the installer pcscfg.cfg are that may be invalid are
 - /usr/lib/gcc/x86_64-redhat-linux/4.1.2/include
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.6/include
 2. Find the pcscfg.cfg file in the correct path in the Linux OS. The path is
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.4
 - /usr/lib/gcc/x86_64-redhat-linux/4.4.7 -> 4.4.4
 3. Back up the pcscfg.cfg file.
 4. Edit the pcscfg.cfg file.
 5. Change the following in the pcscfg.cfg file:


```
/usr/lib/gcc/x86_64-redhat-linux/4.4.6/include to
/usr/lib/gcc/x86_64-redhat-linux/4.4.7/include
```
 6. Run the batch installer.
-
-

(Optional) Analyze Changes in the Patch

Note: See [Appendix: RMS Analyze Tool](#) for details and instructions to run the RMS Analyze Tool. This appendix also contains screens and fields in the tool.

Run the RMS Installer

To run the RMS Installer, complete the following steps:

Note: If Batch is installed along with Database installation then this step can be skipped.

Note: See [Appendix: RMS Batch Installation Screens](#) for details about the RMS Batch installation screens and fields in the installer.

1. Change directories to STAGING_DIR/rms/installer.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc).

Example: prompt\$. oraenv
ORACLE_SID = [] ? mydb
prompt\$

3. Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

Example: prompt\$ echo \$ORACLE_HOME
/u00/oracle/product/mydbversion
prompt\$ echo \$ORACLE_SID
mydb

4. Verify that the following executables are available from PATH: make, makedepend, cc, ar.

Example: Here are some locations where makedepend is commonly found:

Linux:	/usr/bin
SUN:	/usr/bin
AIX:	/usr/bin/X11
HP-UX:	/opt/imate/bin

Set and export the following environment variables. These variables are needed in addition to the environment variables set by the oraenv script above.

Variable	Description	Example
JAVA_HOME	Java home needed to run the GUI. Java 1.8 is required	JAVA_HOME=/usr/java/jdk1.8.64 bit export JAVA_HOME
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERI CA.AL32UTF8 export NLS_LANG
DISPLAY	Address and port of X server on desktop system of user running install. Optional for batch installation	DISPLAY=<IP address>:0 export DISPLAY

5. If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
6. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh [text | silent]
```

7. Verify that the installer reports “SUCCESS” for the Batch preinstall check. If it reports “FAILED,” check for errors in the output under the “Checking environment for Batch installation” section, and verify that your environment variables are set properly.
8. Select the “Patch” option on the Full Install or Patch screen.
9. Check the Install Batch checkbox and continue with installer.
10. On the RETAIL_HOME screen, select the RETAIL_HOME of your previous installation.
11. On the Wallet password screen, enter the wallet password you used in the previous installation.
12. Depending on system resources, a typical RMS batch installation takes around 30 minutes. After the installer is complete, you can check its log file in the “logs” directory: rms-install.<timestamp>.log.
13. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example: `chmod 600 ant.install.properties`

Resolving Errors Encountered During Batch Installation

The RMS batch installation is a full install that starts from the beginning each time it is run. If you encounter errors in your environment, after resolving the issue you can safely run the batch installation again to attempt another installation. Log files for the batch compilation can be found in the RETAIL_HOME/orpatch/logs/rmsbatch/{lib,proc}.

Manual Steps for Running script `ld_iindfiles.ksh`

The prerequisite to using Item Induction is to load the templates on to the database tables. The templates drive the tables, columns to be loaded, and has the translation specific strings.

The below steps are required to load the templates into the environment. This is an optional step and is required only if the client chooses to implement Item Induction functionality.

1. Templates are present in
 `<STAGING_DIR>/rms/installer/mom15/Cross_Pillar/s9t_templates` directory.
2. Review the template and include/exclude the details as required.
3. If not already set, export `TNS_ADMIN=<RETAIL_HOME>/orpatch/rms_wallet`.
4. Go to `<RETAIL_HOME>/oracle/proc/src`.
5. Run `ld_iindfiles.ksh` by passing the two following parameters:
 - `UP=/@<schema owner wallet alias>`
 - Path to folder where the two ods files are located

```
ld_iindfiles.ksh $UP <STAGING_DIR>/rms/installer/mom15/Cross_Pillar/s9t_templates
```

Application Server Installation Tasks – Upgrade

The RMS 15.0.2 installer may be used to upgrade the RMS application. Before you apply the RMS 15.0.2 application upgrade:

- Review the enclosed RMS 15.0.2 Upgrade Release Notes.

Create Staging Directory for RMS Installer

To create the staging directory for the RMS Installer, complete the following steps.

Note: The same installer can be used to install multiple RMS components. If you are installing any of the RMS components (Database, Batch, or Application) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the application server as a user with read and write access to the WebLogic files.
2. Create a staging directory for the RMS installation software.
3. Copy the file `rms15installer.zip` from the RMS 15.0.2 release to the staging directory. This will be referred to as `STAGING_DIR` when installing application software and reports.
4. Change directories to `STAGING_DIR` and extract the file `rms15installer.zip`. This will create an `rms/installer` subdirectory under `STAGING_DIR`.

(Optional) Analyze Changes in the Patch

Note: See [Appendix: RMS Analyze Tool](#) for details and instructions to run the RMS Analyze Tool. This appendix also contains screens and fields in the tool.

Run the RMS Application Installation

Note: See [Appendix: RMS Application Installer Screens](#) for details about the RMS application screens and fields in the installer.

1. Log on to your application server as a user with read and write access to the WebLogic files.
2. Copy the following folders from `ORACLE_HOME`:
 - a. Copy `jdk` folder from `ORACLE_HOME/oracle_common` to `ORACLE_HOME`

```
cp -r /u00/webadmin/products/wls_forms/oracle_common/jdk /u00/webadmin/products/wls_forms
```

- b. Copy ojdbc7.jar from oracle_common/modules/oracle.jdbc/ojdbc7.jar to the ORACLE_HOME/jdbc/lib folder

```
cp
/u00/webadmin/products/wls_forms/oracle_common/modules/oracle.jdbc/ojdbc7.
jar /u00/webadmin/products/wls_forms/jdbc/lib
```

Note: ORACLE_HOME points to the forms installation home

- 3. Change directories to STAGING_DIR/rms/installer.
- 4. Set and export the following environment variables.

Variable	Description	Example
JAVA_HOME	Location of a Java 1. jdk1.8.64bit JDK.	JAVA_HOME= /u00/webadmin/java/jdk1.8.64bit export JAVA_HOME
NLS_LANG	Locale setting for Oracle database client.	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
DOMAIN_HOME	The location where Forms domain has been installed.	DOMAIN_HOME= /u00/webadmin/config/domains/wls_forms/ClassicDomain export DOMAIN_HOME
WLS_INSTANCE	The name of the managed server that contains Oracle Forms.	WLS_INSTANCE=WLS_FORMS export WLS_INSTANCE
ORACLE_HOME	Point to your WebLogic installation	ORACLE_HOME= /u00/webadmin/products/wls_forms export ORACLE_HOME
ORACLE_SID	The database/SID where the RMS schema resides.	ORACLE_SID=mydb export ORACLE_SID
DISPLAY	Address and port of X server on desktop system of user running install. Required for forms application installation.	DISPLAY=<IP address>:0 export DISPLAY

- 5. To install the RMS application you must use an X server (such as Exceed) and have set the DISPLAY environment variable. The installer does not continue otherwise.
- 6. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh [text | silent]
```

Verify that these environment variables are correct. If any of them are incorrect, you need to verify that the WebLogic shell scripts that set them are configured properly. Check the following scripts:

```
$WEBLOGIC_DOMAIN_HOME/bin/setDomainEnv.sh
```

```
$ORACLE_HOME/oracle_common/common/bin/commEnv.sh
```

7. Verify that the installer reports “SUCCESS” for the Application preinstall check. If it reports “FAILED,” check for errors in the output under the “Checking environment for Application installation” section, and verify that your environment variables are set properly.
8. Select the “Patch” option on the Full Install or Patch screen.
9. Check the Install Application checkbox and proceed with the installation.
10. On the RETAIL_HOME screen, select the RETAIL_HOME of your previous installation.
11. On the Wallet password screen, enter the wallet password you used in the previous installation.
12. Depending on system resources, a typical installation can take an hour or more.

Note: You may see the following warning repeated during installation:

```
[exec] Warning! One or more of your selected
locales are not available.
[exec] Please invoke the commands "locale" and
"locale -a" to verify your
[exec] selections and the available locales.
[exec]
[exec] Continuing processing using the "C" locale.

or

[exec] couldn't set locale correctly
```

This warning can be ignored.

13. After the installer is complete, you can check its log file in the “logs” directory:
 STAGING_DIR/rms/installer/logs/rms-install.<timestamp>.log.
 RETAIL_HOME/orpatch/logs/detail_log/{forms/oraforms_rms}
14. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example: `chmod 600 ant.install.properties`

Reports Installation Tasks – Upgrade

RMS Reports are copied to RETAIL_HOME/reports during the application installation.

Installing the RMS BI Publisher Templates

In this section we will outline how the RMS report templates are installed into the appropriate BI server repositories.

Example: <OBIEE_DOMAIN_HOME>/config/bipublisher/repository

Report files are placed by the application installation in the directory - "RETAIL_HOME/reports" and have to be copied into the newly created directory within BI Publisher repository Guest Reports directory.

1. Locate the directory that holds the reports under <BI_REPOSITORY>
Example: <BI_REPOSITORY>/Reports/Guest/RMS
2. Change directory to the RETAIL_HOME/reports/RMS created during the application install. This directory contains subdirectories whose names reflect the names of report templates provided with RMS.
3. Copy each report directory into the directory created above

For example,

```
cp -R * <BI_REPOSITORY>/Reports/Guest/RMS
```

Data Migration

Included in the 15.0 release is a tool responsible for upgrading preexisting data in the RMS schema once 15.0 database upgrades are executed. If upgrading to 15.0, you will need to run this tool to upgrade your data after completing the database patch. Running the tool against schemas that have been patched to a version later than 14.1.1 (for example, 14.1.2) may have unexpected results.

Note: High volume environments may require multiple days for data migration.

Before running the RMS 15.0 Data Migration Tool:

- Make a backup of all your objects and database schema.
- Check that RMS has 15.0 installed.
- Review the RMS 15.0 Release Notes.

Create Staging Directory for RMS Data Migration Files

To create a staging directory for RMS data migration files, complete the following steps.

1. Log in to the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS database schema installation software.
3. Copy the rms15installer.zip file from the RMS 15.0 release to the staging directory. This is referred to as STAGING_DIR when running the data migration tool.
4. Change directories to STAGING_DIR and extract the rms15installer.zip file.

Configure RMS Data Migration Tool

To configure the RMS data migration tool, complete the following steps.

1. Change directories to STAGING_DIR/rms/installer/mom15/Cross_Pillar/upgrade_scripts/source.
2. Create “error”, “log” and “processed” directories.
3. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc).

Example: prompt\$. oraenv
ORACLE_SID = [] ? mydb
prompt\$

4. Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

```
Example: prompt$ echo $ORACLE_HOME
         /u00/oracle/product/mydbversion
prompt$ echo $ORACLE_SID
         mydb
```

5. Set and export the NLS_LANG environment variable.

```
Example:
NLS_LANG=AMERICAN_AMERICA.AL32UTF8
export NLS_LANG
```

6. Set and export the TNS_ADMIN environment variable.

```
Example:
TNS_ADMIN=<RETAIL_HOME>/orpatch/rms_wallet
export TNS_ADMIN
```

7. Open the controller.cfg file and replace the values for the following variables with the appropriate values.

- a. Export PATCH_DIR=
STAGING_DIR/rms/installer/mom15/Cross_Pillar/upgrade_scripts/source
- b. export SCHEMA_OWNER=<The name of the RMS schema>
- c. export MMUSER=/@< Schema Owner Wallet Alias >

```
Note: See Appendix: Setting Up Password Stores with wallets/credential stores for how to set up the database wallet.
```

```
Note: Verify that TNS is set up correctly by using the UP variable to successfully log in to the RMS schema.
```

```
Example: /u00/oracle> sqlplus $UP
```

8. Configure the following files in the STAGING_DIR/rms/installer/mom15/Cross_Pillar/upgrade_scripts/source/files directory with data from your existing RMS schema for the migration. Use the existing files as templates for how this data should be formatted. For descriptions of this data, refer to the RMS 15.0 Data Model document.

- sec_user.dat
sec_user.dat is used to mark the database user as either an RMS user, ReSA User, ReIM user or an Allocation User in the SEC_USER table. Also a manager can be associated to each of the database user. Replace the default values in the template sec_user.dat file with the correct values for your schema. This is an optional upload process.
- sa_user_loc_traits.dat

- `security_config_options.dat`
`security_config_options.dat` is used to update the `DATA_LEVEL_SECURITY_IND` in `SECURITY_CONFIG_OPTIONS` table. Replace the default value in the template `security_config_options.dat` with the correct value for your schema. This is an optional upload process.
- `item_image.dat`
`item_image.dat` is used to upload the image type, primary indicator and display priority for all the images of a given item. Replace the default value in the template `item_image.dat` with the correct value for your schema. This is a mandatory upload process.
- `wh.dat`
`wh.dat` is used to upload the default warehouse for a virtual warehouse in the `WH` table. Replace the default value in the template `wh.dat` with the correct value for your schema. This is an optional upload process.

Run the RMS Data Migration Tool

To run the RMS data migration tool, complete the following steps.

1. Change directories to `STAGING_DIR/rms/installer/mom15/Cross_Pillar/upgrade_scripts/source`.
2. If rerunning the data migration process, clear the contents of the “processed” directory.
3. Run the prevalidation tool. This ensures that the input files for the data migration tool is up to date:

```
$ ./ rms15_upgrade.ksh PREVALIDATION
```
4. Run migration tool.

```
$ ./ rms15_upgrade.ksh UPGRADE
```
5. Run the migration cleanup tool. This removes temporary data migration objects from the database.

```
$ ./ rms15_upgrade.ksh CLEANUP
```
6. Refer to the files in the log and error directory for details if there are problems during migration.
7. You will need to rebuild synonyms for any additional RMS users. Create synonyms to the owner schema for all tables, views, sequences, functions, procedures, packages and types to which the user has access.

Note: The migration tool will also update `ITEM_SUPP_COUNTRY` and `ITEM_SUPP_COUNTRY_LOC` tables to set `NEGOTIATED_ITEM_COST`, `EXTENDED_BASE_COST`, `INCLUSIVE_COST` and `BASE_COST` columns to `NULL` if the default tax type on system options is ‘SVAT’ or ‘SALES’. There are no `.dat` files required for this update as all the records in the tables must be updated.

Run the ReIM Data Migration

To run the ReIM data migration, complete the following steps.

1. Change directories to STAGING_DIR/rms/installer/upgrade_scripts/reim.
2. Log on to sqlplus as the RMS schema owner.
3. Run the following script:

```
@ReIM_Data_Conversion.sql;
```

Run the Allocation Data Migration

To run the Allocation data migration, complete the following steps.

1. Change directories to STAGING_DIR/rms/installer/upgrade_scripts/alloc.
2. Log on to sqlplus as the RMS schema owner.
3. Run the following scripts:

```
@update_alc_location_store_type.sql;  
@update_item_itemdesc_fpg.sql;
```

Web Services Installation

Some Oracle Retail applications; <app> (for example, RMS) use Oracle Objects for the PL/SQL API's. The tool generates a Web Service Provider layer between the external clients and the <app> API's to provide the Web Service functionality, such as faults, logging, and security, as well as the conversion from xml payloads to Oracle Objects. The Retail Service Enabler (RSE) tool creates the appropriate Provider web service endpoints as well as templates for the PL/SQL APIs.

Note: Depending on your business needs, you may not need to install web services.

Set up Environment

To set up the environment, do the following:

1. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc).

Example: prompt\$. oraenv
ORACLE_SID = [] ? mydb
prompt\$

2. Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

Example: prompt\$ echo \$ORACLE_HOME
/u00/oracle/product/mydbversion
prompt\$ echo \$ORACLE_SID
mydb

3. export TNS_ADMIN=/path/to/wallet/files/dir/
4. export UP=/@<Schema Owner Wallet Alias>

Note: See "[Appendix: Setting Up Password Stores with Oracle Wallet](#)" for how to set up database wallet.

5. Verify that TNS is set up correctly by using the UP variable to successfully log in to the RMS 15 schema.

Example: /u00/oracle> sqlplus \$UP

Grant permissions to RMS Database Schema

1. Change directories to STAGING_DIR/rms/installer/mom15/Cross_Pillar/webservice_objects/consumer/sql
2. Change the contents of the following files to your RMS schema owner name when seeing the value <USER>.
 - DrillBackForwardUrlServiceConsumer_grant.sql
 - GLAccountValidationServiceConsumer_grant.sql

Example: Change all occurrence of <USER> to RMS schema owner RMS15DEV in the files:

```
dbms_java.grant_permission( '<USER>',  
'SYS:java.lang.RuntimePermission', 'setFactory', '' )
```

to

```
dbms_java.grant_permission( 'RMS15DEV',  
'SYS:java.lang.RuntimePermission', 'setFactory', '' )
```

Note: For Multitenant databases comment the line CONN / AS SYSDBA)

3. Run the above files as the database SYS user.
4. You do NOT create synonyms to each java object loaded as the synonyms were created in packages previously loaded pointing to the exposed java objects.

Create a Managed Server

Create a managed server for the RMS Web services app to be deployed per the *WebLogic Installation Guide*.

Note: Install WebLogic 12.2.1.2 for Deploying webservice and Restful Service

Create a Datasource

Create a datasource for RMS Webservices to point to the RMS schema as follows:

- Name can be anything you want.
- JNDI Name must be jdbc/RetailWebServiceDs.
- Set database type and driver for your environment (use non-XA jdbc driver).
- Uncheck Supports Global Transactions. Click **Next**
- Set connection properties for the database using the rms user (RMS15DEV). Be sure to test the configuration before moving on.
- Point the data source to the server created in the Create a Managed Server section above.

Deploy RMS Service EAR File

To deploy the RMS Service .ear file, do the following.

1. Make sure that the managed server created in Step 2, where this application will be deployed, is up and running.
2. In the left Domain Structure window, click Environment > Deployments.

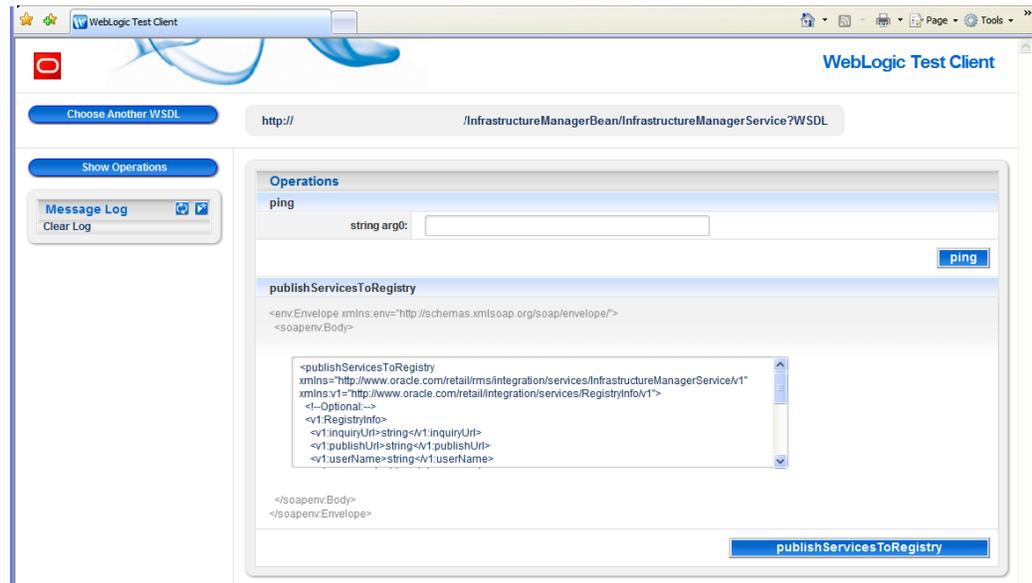
3. Click **Lock and Edit** in the change center to install the ear file. It will enable the install button on the deployments screen.
4. Click **Install**.
5. Click the **upload your file(s)** link.
6. Click the **Deployment Archive** browse button.
7. Select the `rms-service.ear` file from local machine This file can be found in: `STAGING_DIR/rms/installer/mom15/Cross_Pillar/webservice_objects/provider/ear`.
8. Click **Next**. Make sure that the radio button for `rms-services.ear` is selected.
9. Click **Next** again. Make sure that **Install this deployment as an application** is selected.
10. Click **Next** again and select the server created in Step 2.
11. Click **Next**. Click **Finish** to return to the deployments page. You should see `rms-service` in the list of deployments.
12. Click **Activate Changes** in the change center. The state of the application may be shown as prepared. If so, select the check box next to `rms-service` to will enable the **Start** button. Click **Start**. Select servicing all requests.
13. To test the deployment, click on the application. Click the **Testing** tab.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled 'Settings for rms-service' and has the 'Testing' tab selected. Below the tabs, there is a table of 'Deployment Tests' for the 'rms-service' application. The table has three columns: 'Name', 'Test Point', and 'Comments'. The table lists various test points for different modules and services within the application.

Name	Test Point	Comments
InfrastructureManagerBean		There are no test points for this EJB module.
InfrastructureManagerService		Test points for this WebService module.
/InfrastructureManagerBean/InfrastructureManagerService	?WSDL	WSDL page on server rms-service-server
/InfrastructureManagerBean/InfrastructureManagerService	Test client	Test client on server rms-service-server
PayTermBean		There are no test points for this EJB module.
PayTermService		Test points for this WebService module.
ReportLocatorBean		There are no test points for this EJB module.
ReportLocatorService		Test points for this WebService module.
SupplierBean		There are no test points for this EJB module.
SupplierService		Test points for this WebService module.

The console also shows a 'Change Center' on the left with 'Lock & Edit' and 'Release Configuration' buttons. The 'System Status' section at the bottom left shows the health of running servers as 'OK (15)'. The footer of the console displays 'WebLogic Server Version: 10.3.3.0' and copyright information.

14. Expand one of the four web services. Click the ?WSDL and Test Client links to test. For the test client you should see a screen similar to the following:



Note: If Webservice ear is being deployed in a managed server which is running on HTTPS port, the Test Point WSDL links may not be visible. In order to view the WSDL links, you need to temporary enable the HTTP (non-secured) ports of Administration and the webservice managed server and login and navigate to deployments of webservice to view the WSDL links.

Configure Web Service Security

Configuring the web service deployment to use the WS-Security Username Profile involves forcing all incoming requests to contain WS Security headers to authenticate the requestor based on a user name and password elements. The use of this profile does not provide any confidentiality protection on web service requests: data contained within the Web service messages will not be encrypted. However, using a secure message transport, such as SSL/TLS, will provide confidentiality for the message as it traverses the network. For more information on using SSL/TLS see the section, “Configuring SSL” found in the WebLogic document, “Securing the WebLogic Server, 10g Release 3 (10.3)”.

Additional WS Security policies may also be available depending on the configuration of the WebLogic server. Using these policies will require appropriate changes to web service requests created by applications consuming the web service. Many of these policies also require additional steps for correct keystore and truststore file configuration.

When a web service uses the WS-Security Username profile, all web service consumers must specify a user name configured within the current WebLogic domain. This user name must also have the appropriate role(s) associated with it. Using this profile is thus a two-step process:

1. Attach the WS-Security Username policy to the web service
2. Create roles and users who can access the web services

Refer to *Oracle Retail Merchandising Security Guide* along with *Oracle Retail Service Backbone Security Guide* for more details.

Configuring RMS RESTful Web Services:

Create a Managed Server

Set up the environment as above mentioned in Web Services Installation and configure managed server for deploying the mobile services for RMS. Create a managed server for the RMS Web services app to be deployed per the *WebLogic Installation Guide*.

Note: Install WebLogic 12.2.1.2 for deploying RestFul Service and while configuring the managed server it should be extended with ADF libraries.

Create a Datasource

Create a datasource for RMS Webservices to point to the RMS schema as follows.

- Name can be anything you want.
- JNDI Name must be jdbc/RMSDBDS.
- Set database type and driver for your environment (use non-XA jdbc driver).
- Uncheck Supports Global Transactions. Click **Next**
- Set connection properties for the database using the rms user (RMS15DEV). Be sure to test the configuration before moving on.
- Point the data source to the server created in the Create a Managed Server section above.

Deploy RMS RESTful Web Service File

Perform the following procedure to deploy the EarRMSService.ear file:

1. Make sure that the managed server created in Step 2, where this application will be deployed, is up and running.
2. In the left Domain Structure window, click Environment > Deployments.
3. Click Lock and Edit in the change center to install the ear file. It will enable the install button on the deployments screen.
4. Click **Install**.
5. Click the upload your file(s) link.
6. Click the Deployment Archive browse button.
7. Select the EarRMSService.ear file from local machine.
8. Click **Next**. Make sure that the radio button for EarRMSService.ear is selected.
9. Click **Next** again. Make sure that Install this deployment as an application is selected.
10. Click **Next** again and select the server created in Step 2.
11. Click **Next**. Click **Finish** to return to the deployments page. You should see **EarRMSService** in the list of deployments.
12. Click **Activate Changes** in the change center.
13. To test the deployment, click on the application. Click the testing tab and test the deployed url.

Example URL: <https://<hostname>:<Managed port>/RMSService>

Configure Users and Group:

For external Security, the role must be created in the LDAP and sourced to the WebLogic Security Realm

- Go to admin console where RMS Mobile or REST service is deployed.
- Go to security realm -> myrealm ->Users and Groups -> Groups, then click the New button and add a group called DATA_STEWARD_MANAGER_JOB.
- Then add users and add the above configured group to this user. The users should have the DATA_STEWARD_MANAGER_JOB group.

You can now login to the application and the REST WSDL with the users which are under the DATA_STEWARD_MANAGER_JOB Group

Note: Login to the webservice url with the above configured users and group.

Example URL:

<http://<Hostname>:<Portno>/RMSService/services/private/PurchaseOrders/>

Patching Procedures

Oracle Retail Patching Process

The patching process for many Oracle Retail products has been substantially revised from prior releases. Automated tools are available to reduce the amount of manual steps when applying patches. To support and complement this automation, more information about the environment is now tracked and retained between patches. This information is used to allow subsequent patches to identify and skip changes which have already been made to the environment. For example, the patching process uses a database manifest table to skip database change scripts which have already been executed.

The enhanced product patching process incorporates the following:

- Utilities to automate the application of Oracle Retail patches to environments.
- Unified patches so that a single patch can be applied against Database, Forms, Java applications, Batch, etc. installations.
- Database and Environment manifests track versions of files at a module level.
- Centralized configuration distinguishes installation types (Database, Forms, Java, Batch, etc.).
- Patch inventory tracks the patches applied to an environment.

These enhancements make installing and updating Oracle Retail product installations easier and reduce opportunities for mistakes. Some of these changes add additional considerations to patching and maintaining Oracle Retail product environments. Additional details on these considerations are found in later sections.

Supported Products and Technologies

With version 15.0, several products and technologies are supported by the enhanced patching process. The utilities, processes and procedures described here are supported with the following products and listed technologies:

Product	Supported Technology
Oracle Retail Merchandising System (RMS)	<ul style="list-style-type: none"> ▪ Database scripts ▪ Batch scripts ▪ RETL scripts ▪ Data Conversion Scripts ▪ Forms ▪ BI Publisher Reports
Oracle Retail Warehouse Management System (RWMS)	<ul style="list-style-type: none"> ▪ Database scripts ▪ Batch scripts ▪ Forms ▪ BI Publisher Reports

Product	Supported Technology
Oracle Retail Price Management (RPM)	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application ▪ Batch scripts
Oracle Retail Invoice Matching (ReIM)	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application ▪ Batch scripts
Oracle Retail Allocation	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application ▪ Batch scripts
Oracle Retail Sales Audit (ReSA)	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application
Oracle Retail Insights (RI) Previously called Oracle Retail Analytics (RA)	<ul style="list-style-type: none"> ▪ Database scripts
Oracle Retail Advanced Science Engine (ORASE)	<ul style="list-style-type: none"> ▪ Database scripts ▪ Batch scripts
Oracle Retail Data Extractor (RDE)	<ul style="list-style-type: none"> ▪ Database scripts
Oracle Retail Application Admin Console (ORAAC). Previously called Oracle Retail Application Security Role Manager (RASRM)	<ul style="list-style-type: none"> ▪ Java Application

Patch Concepts

During the lifecycle of an Oracle Retail environment, patches are applied to maintain your system. This maintenance may be necessary to resolve a specific issue, add new functionality, update to the latest patch level, add support for new technologies, or other reasons.

A patch refers to a collection of files to apply to an environment. Patches could be cumulative, such as the 15.0.2 release, or incremental, such as a hot fix for just a few modules. Patches may contain updates for some or all components of a product installation including database, application code, forms, and batch. In a distributed architecture the same patch may need to be applied to multiple systems in order to patch all of the components. For example, if a patch contains both database and application changes, the patch would need to be applied to both the database server and the application server.

The top-level directory for the installation of an Oracle Retail product is referred to as the `RETAIL_HOME`. Underneath `RETAIL_HOME` are all of the files related to that product installation, as well as configuration and metadata necessary for the Oracle Retail Patch Assistant to maintain those files. In some cases the runtime application files also exist under `RETAIL_HOME`. For example, the compiled RMS forms, compiled RMS batch files, or Java Application batch scripts.

Patching Utility Overview

Patches are applied and tracked using utilities that are specifically designed for this purpose. The primary utility is described briefly below and additional information is available in later sections.

Oracle Retail Patch Assistant (ORPatch)

ORPatch is the utility used to apply patches to an Oracle Retail product installation. It is used in the background by the installer when creating a new installation or applying a cumulative patch. It is used directly to apply an incremental patch to an environment.

Oracle Retail Merge Patch (ORMerge)

ORMerge is a utility to allow multiple patches to be combined into a single patch. Applying patches individually may require some steps to be repeated. Merging multiple patches together allows these steps to be run only once. For example, applying several incremental patches to database packages will recompile invalid objects with each patch. Merging the patches into a single patch before applying them will allow invalid objects to be recompiled only once.

Oracle Retail Compile Patch (ORCompile)

ORCompile is a utility to compile components of Oracle Retail products outside of a patch. It allows RMS Forms, RMS Batch, and RWMS Forms to be fully recompiled even if no patch has been applied. It also contains functionality to recompile invalid database objects in product schemas.

Oracle Retail Deploy Patch (ORDeploy)

ORDeploy is a utility to deploy components of Oracle Retail Java products outside of a patch. It allows RPM, ReIM, Allocation and ReSA java applications to be redeployed to WebLogic even if a patch has not been applied. It contains functionality to optionally include or not include Java customizations when redeploying.

Changes with 15.0

Some products and technologies are supported by the enhanced patching process for the first time in 15.0. In those cases all of the content in this chapter is new with 15.0.

Renamed Products

For the 15.0 release several products using Orpatch have been renamed. Oracle Retail Analytics (RA) is renamed to Oracle Retail Insights (RI) and Oracle Retail Application Security Role Manager (RASRM) is renamed to Oracle Retail Application Admin Console (ORAAC). The old product names are still used in the underlying Orpatch code, the Orpatch Action names, the Orpatch env_info.cfg, and the Orpatch wallet paths. Documentation for these products will refer to the new names.

Patching Considerations

Patch Types

Oracle Retail produces two types of patches for their products: cumulative and incremental.

Cumulative Patches

A cumulative patch includes all of the files necessary to patch an environment to a specific level or build a new environment at that level. Examples of cumulative patches would be 15.0.2, 15.0.2, and so on. Cumulative patches come with a standard Oracle Retail installer and so can be applied to an environment with the installer rather than with ORPatch or other utilities.

Incremental Patches

An incremental patch includes only selected files necessary to address a specific issue or add a feature. Examples of incremental patches would be a hot fix for a specific defect. Incremental patches do not include an installer and must be applied with ORPatch.

Incremental Patch Structure

An Oracle Retail incremental patch generally contains several files and one or more subdirectories. The subdirectories contain the contents of the patch, while the individual files contain information about the patch and metadata necessary for patching utilities to correctly apply the patch. The most important files in the top-level directory are the README.txt, the manifest files.

README File

The README.txt file contains information about the incremental patch and how to apply it. This may include manual steps that are necessary before, after or while applying the patch. It will also contain instructions on applying the patch with ORPatch.

Manifest Files

Each patch contains manifest files which contain metadata about the contents of a patch and are used by ORPatch to determine the actions necessary to apply a patch. Patches should generally be run against all installations a product in an environment, and ORPatch will only apply the changes from the patch that are relevant to that installation.

Note: Cumulative patches use a different patch structure because they include a full installer which will run ORPatch automatically.

Version Tracking

The patching infrastructure for 15.0 tracks version information for all files involved with a product installation. The RETAIL_HOME contains files which track the revision of all files within the RETAIL_HOME including batch, forms, database, Java archives and other files. In addition, records of database scripts that have been applied to the product database objects are kept within each database schema.

Apply all Patches with Installer or ORPatch

In order to ensure that environment metadata is accurate all patches must be applied to the Oracle Retail product installation using patching utilities. For cumulative patches this is done automatically by the installer. For incremental patches ORPatch must be used directly. This is especially important if database changes are being applied, in order to ensure that the database-related metadata is kept up-to-date.

Environment Configuration

A configuration file in `$RETAIL_HOME/orpatch/config/env_info.cfg` is used to define the details of a specific Oracle Retail environment. This file defines:

- The location of critical infrastructure components such as the `ORACLE_HOME` on a database or middleware server.
- The location of Oracle Wallets to support connecting to the database users.
- The type of file processing which is relevant to a particular host. For example, if this is a host where database work should be done, or a host where batch compilation should be done, a host where Java applications should be deployed, etc. This allows a single database, forms and batch patch to be run against all types of hosts, applying only the relevant pieces on each server.
- Other configuration necessary to determine proper behavior in an environment.

Retained Installation Files

The `RETAIL_HOME` location of an Oracle Retail product installation contains all of the files associated with that installation. This can include database scripts, Java files, Forms, Batch, RETL and Data Conversion files as with previous versions and also includes all database scripts. This allows objects to be reloaded during patching, including any necessary dependencies.

Reloading Content

In order to ensure that database contents and generated files exactly match patched versions, when applying cumulative patches some content is regenerated even if it does not appear to have changed.

On a cumulative patch this includes:

- All re-runnable database content will be reloaded
 - Packages and Procedures
 - Database Types (excluding RIB objects)
 - Control scripts
 - Triggers
 - Webservice jars and packages
 - Form Elements
- All RMS and RWMS forms files will be recompiled
- All RMS batch files will be recompiled

When applying incremental patches, only changed files will be reloaded. However this does not apply to RMS batch, which is fully recompiled with any change.

Java Hotfixes and Cumulative Patches

When applying cumulative patches to Java applications components with `ORPatch`, all hotfixes related to base product ear files included with the patch will be rolled back. This increases the likelihood of a successful deployment because hotfixes may not be compatible with updated product ear files, or may already be included with the ear. Before applying a cumulative patch to Java applications, check the patch documentation to determine which hotfixes are not included in the ear. Then work with Oracle Support to obtain compatible versions of the fixes for the updated ear version. In some cases this may be the same hotfix, in which case it can be re-applied to the environment. In other cases a new hotfix may be required.

Backups

Before applying a patch to an environment, it is extremely important to take a full backup of both the RETAIL_HOME file system and the Oracle Retail database. Although ORPatch makes backups of files modified during patching, any database changes cannot be reversed. If a patch fails which contains database changes, and cannot be completed, the environment must be restored from backup.

Disk Space

When patches are applied to an environment, the old version of files which are updated or deleted are backed up to \$RETAIL_HOME/backups/backup-<timestamp>. When applying large patches, ensure there is sufficient disk space on the system where you unzip the patch or the patching process may fail. Up to twice as much disk space as the unzipped patch may be required during patching.

In addition to backups of source files, the existing compiled RMS or RWMS Forms and RMS Batch files are saved before recompilation. These backups may be created during patches:

- Batch 'lib' directory in \$RETAIL_HOME/oracle/lib/bin-<timestamp>
- Batch 'proc' directory in \$RETAIL_HOME/oracle/proc/bin-<timestamp>
- Forms 'toolset' directory in \$RETAIL_HOME/base/toolset/bin-<timestamp>
- Forms 'forms' directory in \$RETAIL_HOME/base/forms/bin-<timestamp>

Periodically both types of backup files can be removed to preserve disk space.

Patching Operations

Running ORPatch

ORPatch is used to apply patches to an Oracle Retail product installation. When applying a patch which includes an installer, ORPatch does not need to be executed manually as the installer will run it automatically as part of the installation process. When applying a patch that does not include an installer, ORPatch is run directly.

ORPatch performs the tasks necessary to apply the patch:

- Inspects the patch metadata to determine the patch contents and patch type.
- Reads the environment configuration file to determine which product components exist in this installation.
- Assembles a list of patch actions which will be run on this host to process the patch.
- Executes pre-checks to validate that all patch actions have the necessary configuration to proceed.
- Compares version numbers of files from the patch against the files in the environment.
- Backs up files which will be updated.
- Copies updated files into the installation.
- Loads updated files into database schemas, if applicable.
- Recompiles RMS batch, if applicable.
- Recompiles RMS forms, if applicable.
- Recompiles RWMS forms, if applicable.
- Constructs updated Java archives and deploys them to WebLogic, if applicable
- Updates Java batch files and libraries, if applicable
- Records the patch in the patch inventory.

If a patch does not contain updated files for the database or system, no action may be taken. If a previously failed ORPatch session is discovered, it will be restarted.

Preparing for Patching

Before applying a patch to your system, it is important to properly prepare the environment.

Single Patching Session

It is extremely important that only a single ORPatch session is active against a product installation at a time. If multiple patches need to be applied, you can optionally merge them into a single patch and apply one patch to the environment. Never apply multiple patches at the same time.

Shutdown Applications

If a patch updates database objects, it is important that all applications are shutdown to ensure no database objects are locked or in use. This is especially important when applying changes to Oracle Retail Integration Bus (RIB) objects as types in use will not be correctly replaced, leading to “ORA-21700: object does not exist or marked for delete” errors when restarting the RIB.

Backup Environment

Before applying a patch to an environment, it is important to take a full backup of both the RETAIL_HOME file system and the retail database. Although ORPatch makes backups of files modified during patching, any database changes cannot be reversed. If a patch which contains database changes fails and cannot be completed, the environment must be restored from backup.

Log Files

When applying a patch, ORPatch will create a number of log files which contain important information about the actions taken during a patch and may contain more information in the event of problems. Log files are created in the \$RETAIL_HOME/orpatch/logs directory. Logs should always be reviewed after a patch is applied.

After a patch session the log directory will contain at a minimum an ORPatch log file and may also contain other logs depending on the actions taken. The following table describes logs that may exist.

Log File	Used For
orpatch-<date>-<time>.log	Primary ORPatch log file
detail_logs/dbsql_<component>/invalids/*	Details on the errors causing a database object to be invalid
detail_logs/analyze/details	Detail logs of files that will be created/updated/removed when a patch is applied
detail_logs/compare/details	Detail logs of the differences between two sets of environment metadata
orpatch_forms_<pid>_child_<num>.log	Temporary logs from a child process spawned to compile forms in parallel. After the child process completes, the contents are append to the primary orpatch log file
detail_logs/forms/rms_frm_toolset/*	Detail logs of the compilation of each RMS Toolset file
detail_logs/forms/rms_frm_forms/*	Detail logs of the compilation of each RMS Forms file
detail_logs/rmsbatch/lib/*	Detail logs of the compilation of RMS Batch libraries
detail_logs/rmsbatch/proc/*	Detail logs of the compilation of RMS Batch programs
detail_logs/dbsql_rms/rms_db_ws_consumer_jars/*	Detail logs of the loadjava command to install RMS WebService Consumer objects
detail_logs/dbsql_rms/rms_db_ws_consumer_libs/*	Detail logs of the loadjava command to install RMS WebService Consumer libraries

Log File	Used For
detail_logs/forms/rwms_frm_forms/*	Detail logs of the compilation of each RWMS Forms file
detail_logs/dbsql_rwms/rwms_db_sp_jars/*	Detail logs of the loadjava command to install RWMS SP jars
detail_logs/javaapp_<product>/deploy/*	Detail logs of the deploy of a Java product

Unzip Patch Files

Before executing ORPatch, the patch files must be unzipped into a directory. This directory will be passed to ORPatch as the “-s <source directory>” argument on the command-line when applying or analyzing a patch.

Location of ORPatch

The ORPatch script will be located in \$RETAIL_HOME/orpatch/bin.

Command Line Arguments

ORPatch behavior is controlled by several command-line arguments. These arguments may be actions or options. Command and option names can be specified in upper or lower case, and will be converted to upper-case automatically. Arguments to options, for example the source directory patch, will not be modified.

ORPatch command-line actions:

Action	Description
apply	Tells ORPatch to apply a patch, requires the -s option Example: orpatch apply -s \$RETAIL_HOME/stage/patch123456
analyze	Tells ORPatch to analyze a patch, requires the -s option Example: orpatch analyze -s \$RETAIL_HOME/stage/patch123456
lsinventory	Tells ORPatch to list the inventory of patches that have been applied to this installation
exportmetadata	Tells ORPatch to extract all metadata information from the environment and create a \$RETAIL_HOME/support directory to contain it. Requires the -expname option.
diffmetadata	Tells ORPatch to compare all metadata from the current environment with metadata exported from some other environment. Requires the -expname and -srname options.
revert	Tells ORPatch to revert the files related to a patch, requires the -s option Example: orpatch revert -s \$RETAIL_HOME/backups/backup-09302013-153010

Note: An action is required and only one action can be specified at a time.

ORPatch command-line arguments:

Argument	Valid For Actions	Description
-s <source dir>	apply analyze	Specifies where to find the top-level directory of the patch to apply or analyze. The source directory should contain the manifest.csv and patch_info.cfg files.
-new	apply	Forces ORPatch to not attempt to restart a failed ORPatch session
-expname	exportmetadata diffmetadata lsinventory	Defines the top-level name to be used for the export or comparison of environment metadata. When used with lsinventory, it allows an exported inventory to be printed.
-srcname	diffmetadata	Defines the 'name' to use when referring to the current environment during metadata comparisons.
-dbmodules	diffmetadata	When comparing metadata at a module-level, compare the dbmanifest information rather than the environment manifest. This method of comparing metadata is less accurate as it does not include non-database files.
-jarmodules	analyze diffmetadata	When used with analyze, requests a full comparison of the metadata of Java archives included in the patch versus the metadata of the Java archives in the environment. This behavior is automatically enabled when Java customizations are detected in the environment. Analyzing the contents of Java archives allows for detailed investigation of the potential impacts of installing a new Java ear to an environment with customizations. When used with diffmetadata, causes metadata to be compared using jarmanifest information rather than the environment manifest. This provides more detailed information on the exact differences of the content of Java archives, but does not include non-Java files.
-selfonly	apply analyze	Only apply or analyze changes in a patch that relate to orpatch itself. This is useful for applying updates to orpatch without applying the entire patch to an environment.
-s <backup dir>	revert	Specifies the backup from a patch that should be reverted to the environment. This restores only the files modified during the patch, the database must be restored separately or the environment will be out-of-sync and likely unusable.

Analyzing the Impact of a Patch

In some cases, it may be desirable to see a list of the files that will be updated by a patch, particularly if files in the environment have been customized. ORPatch has an 'analyze' mode that will evaluate all files in the patch against the environment and report on the files that will be updated based on the patch.

To run ORPatch in analyze mode, include 'analyze' on the command line. It performs the following actions:

- Identifies files in the environment which the patch would remove.
- Compares version numbers of files in the patch to version numbers of files in the environment.
- Prints a summary of the number of files which would be created, updated or removed.
- Prints an additional list of any files that would be updated which are registered as being customized.
- Prints an additional list of any files which are in the environment and newer than the files included in the patch. These files are considered possible conflicts as the modules in the patch may not be compatible with the newer versions already installed. If you choose to apply the patch the newer versions of modules in the environment will NOT be overwritten.
- If a Java custom file tree is detected, prints a detailed analysis of the modules within Java ear files that differ from the current ear file on the system.
- Saves details of the files that will be impacted in `$RETAIL_HOME/orpatch/logs/detail_logs/analyze/details`.

This list of files can then be used to assess the impact of a patch on your environment.

To analyze a patch, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Set the JAVA_HOME environment variable if the patch contains Java application files.

```
export JAVA_HOME=/u00/oretail/java_jdk
```

Note: If the JAVA_HOME environment variable is not specified, the value from `RETAIL_HOME/orpatch/config/env_info.cfg` will be used.

5. Create a staging directory to contain the patch, if it does not already exist.

```
mkdir -p $RETAIL_HOME/stage
```
6. Download the patch to the staging directory and unzip it.
7. Execute orpatch to analyze the patch.

```
orpatch analyze -s $RETAIL_HOME/stage/patch123456
```
8. Repeat the patch analysis on all servers with installations for this product environment.
9. Evaluate the list(s) of impacted files.

For more information on registering and analyzing customizations, please see the Customization section later in this document.

Applying a Patch

Once the system is prepared for patching, ORPatch can be executed to apply the patch to the environment. The patch may need to be applied to multiple systems if it updates components that are installed on distributed servers.

To apply a patch, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```

3. Set the PATH environment variable to include the orpatch/bin directory
4. Set the DISPLAY environment variable if the patch contains Forms.

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```

```
export DISPLAY=localhost:10.0
```

Note: If the DISPLAY environment variable is not specified, the value from RETAIL_HOME/orpatch/config/env_info.cfg will be used.

5. Set the JAVA_HOME environment variable if the patch contains Java application files.

```
export JAVA_HOME=/u00/oretail/java_jdk
```

Note: If the JAVA_HOME environment variable is not specified, the value from RETAIL_HOME/orpatch/config/env_info.cfg will be used.

6. Create a staging directory to contain the patch, if it does not already exist.
7. Download the patch to the staging directory and unzip it.
8. Review the README.txt included with the patch. If manual steps are specified in the patch, execute those steps at the appropriate time.
9. Shutdown applications.
10. Execute ORPatch to apply the patch.
11. After ORPatch completes, review the log files in \$RETAIL_HOME/orpatch/logs.
12. Repeat the patch application on all servers with installations for this product environment.
13. Restart applications.

```
mkdir -p $RETAIL_HOME/stage
```

```
orpatch apply -s $RETAIL_HOME/stage/patch123456
```

Restarting ORPatch

If ORPatch is interrupted while applying a patch, or exits with an error, it saves a record of completed work in a restart state file in \$RETAIL_HOME/orpatch/logs. Investigate and resolve the problem that caused the failure, then restart ORPatch.

By default when ORPatch is started again, it will restart the patch process close to where it left off. If the patch process should **not** be restarted, add '-new' to the command-line of ORPatch.

Please note that starting a new patch session without completing the prior patch may have serious impacts that result in a patch not being applied correctly. For example, if a patch contains database updates and batch file changes and ORPatch is aborted during the load of database objects, abandoning the patch session will leave batch without the latest changes compiled in the installation.

Listing the Patch Inventory

After a patch is successfully applied by ORPatch the patch inventory in `$RETAIL_HOME/orphatch/inventory` is updated with a record that the patch was applied. This inventory contains a record of the patches applied, the dates they were applied, the patch type and products impacted.

To list the patch inventory, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the `RETAIL_HOME` environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```

3. Set the `PATH` environment variable to include the `orphatch/bin` directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```

4. Execute `orphatch` to list the inventory.

```
orphatch lsinventory
```

Exporting Environment Metadata

ORPatch functionality is driven based on additional metadata that is stored in the environment to define what version of files are applied to the environment, and which database scripts have been applied to database schemas. This environment metadata is used to analyze the impact of patches to environments and controls what actions are taken during a patch. The metadata is stored in several locations depending on the type of information it tracks and in some cases it may be desirable to extract the metadata for analysis outside of ORPatch. For example, Oracle Support could ask for the metadata to be uploaded to assist them in triaging an application problem.

ORPatch provides a capability to export all of the metadata in an environment into a single directory and to automatically create a zip file of that content for upload or transfer to another system. The exact metadata collected from the environment depends on the products installed in the `RETAIL_HOME`.

ORPatch metadata exported:

Installed Product Component	Exported Metadata	Description
Any	<code>orphatch/config/env_info.cfg</code> <code>orphatch/config/custom_hooks.cfg</code> ORPatch inventory files	ORPatch configuration and settings
Any	All <code>env_manifest.csv</code> and <code>deleted_env_manifest.csv</code> files	Environment manifest files detailing product files installed, versions, customized flags and which patch provided the file
Database Schemas	DBMANIFEST table contents	Database manifest information detailing which database scripts were run, what version and when they were executed
Java Applications	All files from <code>javaapp_<product>/config</code> except jar files	Environment-specific product configuration files generated during installation

Installed Product Component	Exported Metadata	Description
Java Applications	Combined export of all META-INF/env_manifest.csv files from all product ear files	Jar manifest information detailing files, versions, customized flags and which patch provided the file
Java Applications	orpatch/config/javaapp_<product>/ant.deploy.properties	Environment properties file created during product installation and used during application deployment
Java Applications	<weblogic_home>/server/lib/weblogic.policy	WebLogic server java security manager policy file
RMS Batch	orpatch/config/rmsbatch_profile	Batch compilation shell profile
RMS Forms	orpatch/config/rmsforms_profile	Forms compilation shell profile
RWMS Forms	orpatch/cofngi/rwsmforms_profile	Forms compilation shell profile

Exports of environment metadata are always done to the \$RETAIL_HOME/support directory. When exporting metadata, you must specify the `-exname` argument and define the name that should be given to the export. The name is used for the directory within \$RETAIL_HOME/support and for the name of the zip file.

To extract an environment's metadata, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Execute orpatch to export the metadata.

```
orpatch exportmetadata -exname test_env
```

This example would export all metadata from the environment to the \$RETAIL_HOME/support/test_env directory. A zip file of the metadata would be created in \$RETAIL_HOME/support/test_env.zip.

Note: The \$RETAIL_HOME/support/<name> directory should be empty or not exist prior to running `exportmetadata` in order to ensure accurate results.

Comparing Environment Metadata

Once metadata has been exported from an environment, it can be used to compare the environment manifest metadata of two environments. ORPatch provides a capability to compare metadata of the current environment with the exported metadata of another environment. Note that even though there are many types of metadata exported by ORPatch, only environment manifest metadata is evaluated during comparisons. Metadata comparison happens in four phases: product comparison, patch comparison, ORPatch action comparison, and module-level comparison.

Product comparison compares the products installed in one environment with the products installed in another environment. Patch comparison compares the patches applied in one environment with the patches applied in another environment, for common products. This provides the most summarized view of how environments

differ. Patches which only apply to products on one environment are not included in the comparison.

Since each patch may impact many files, the comparison then moves on to more detailed analysis. The third phase of comparison is to compare the enabled ORPatch actions between environments. These actions roughly correspond to the installed 'components' of a product. For example, one environment may have database and forms components installed while another has only forms. Action comparison identifies components that are different between environments. The final phase of comparison is at the module level for actions that are common between environments. Modules which exist only on one environment, or exist on both environments with different revisions, or which are flagged as customized are reported during the comparison.

Differences between environment metadata are reported in a summarized fashion during the ORPatch execution. Details of the comparison results are saved in `$RETAIL_HOME/orphatch/logs/detail_logs/compare/details`. One CSV file is created for each phase of comparison: `product_details.csv`, `patch_details.csv`, `action_details.csv` and `module_details.csv`.

In order to be compared by ORPatch, exported metadata must be placed in the `$RETAIL_HOME/support` directory. The metadata should exist in the same structure that it was originally exported in. For example, if the metadata was exported to `$RETAIL_HOME/support/test_env` on another system, it should be placed in `$RETAIL_HOME/support/test_env` on this system.

When reporting differences between two environments, ORPatch uses names to refer to the environments. These names are defined as part of the `diffmetadata` command. The `-expname` parameter, which defines the directory containing the metadata, is also used as the name when referring to the exported metadata. The `-srcname` parameter defines the name to use when referring to the current environment. As an example, if you had exported the 'test' environment's metadata and copied it to the 'dev' environment's `$RETAIL_HOME/support/test_env` directory, you could run "`orphatch diffmetadata -expname test_env -srcname dev_env`". The detail and summary output would then refer to things that exist on dev but not test, revisions in the test environment versus revisions in the dev environment, etc.

ORPatch will automatically export the environment's current metadata to `$RETAIL_HOME/support/compare` prior to starting the metadata comparison.

To compare two environment's metadata, perform the following steps:

1. Export the metadata from another environment using `orphatch exportmetadata`.
2. Transfer the metadata zip from the other system to `$RETAIL_HOME/support`.
3. Log in as the UNIX user that owns the product installation.
4. Set the `RETAIL_HOME` environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/dev
```

5. Set the `PATH` environment variable to include the `orphatch/bin` directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```

6. Unzip the metadata zip file.

```
unzip test_env.zip
```

7. Execute `orphatch` to compare the metadata

```
orphatch diffmetadata -expname test_env -srcname dev_env
```

This example would compare the current environment against the metadata extracted in `$RETAIL_HOME/support/test_env` directory.

Note: The \$RETAIL_HOME/support/compare directory will be automatically removed before environment metadata is exported at the start of the comparison.

Reverting a Patch

In general it is best to either completely apply a patch, or restore the entire environment from the backup taken before starting the patch. It is important to test patches in test or staging environments before applying to production. In the event of problems, Oracle Retail recommends restoring the environment from backup if a patch is not successful.

Note: Reverting patches in an integrated environment can be extremely complex and there is no fully automated way to revert all changes made by a patch. Restoring the environment from a backup is the recommended method to remove patches.

It is, however, possible to revert small patches using the backups taken by ORPatch during a patch. This will restore only the files modified, and it is still necessary to restore the database if any changes were made to it.

Note: Reverting a patch reverts only the files modified by the patch, and does not modify the database, or recompile forms or batch files after the change.

When multiple patches have been applied to an environment, reverting any patches other than the most recently applied patch is strongly discouraged as this will lead to incompatible or inconsistent versions of modules applied to the environment. If multiple patches are going to be applied sequentially it is recommended to first merge the patches into a single patch that can be applied or reverted in a single operation.

To revert a patch, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Identify the backup directory in \$RETAIL_HOME/backups that contains the backup from the patch you want to restore.
 - The backup directory will contain a patch_info.cfg file which contains the name of the patch the backup is from.
 - It is possible to have two directories for the same patch, if ORPatch was updated during the patch. It is not possible to revert the updates to ORPatch. Select the backup directory that does not contain orpatch files.
 - If it is not clear which backup directory to use, restore the environment from backup
5. Execute orpatch to revert the environment using the contents of the backup directory

```
orpatch revert -s $RETAIL_HOME/backups/backup-11232013-152059
```
6. Restore the database from backup if the patch made database changes
7. Use the orcompile script to recompile forms if the patch included RMS or RWMS forms files

```
orcompile -a RMS -t FORMS
orcompile -a RWMS -t FORMS
```

8. Use the orcompile script to recompile batch if the patch included RMS batch files
orcompile -a RMS -t BATCH

9. Use the ordeploy script to redeploy the appropriate Java applications if the patch included Java files

```
ordeploy -a RPM -t JAVA
ordeploy -a REIM -t JAVA
ordeploy -a ALLOC -t JAVA
ordeploy -a RESA -t JAVA
```

Merging Patches

When patches are applied individually some ORPatch tasks such as compiling forms and batch files or deploying Java archives are performed separately for each patch. This can be time-consuming. An alternative is to use the ORMerge utility to combine several patches into a single patch, reducing application downtime by eliminating tasks that would otherwise be performed multiple times. Patches merged with ORMerge are applied with ORPatch after the merge patch is created.

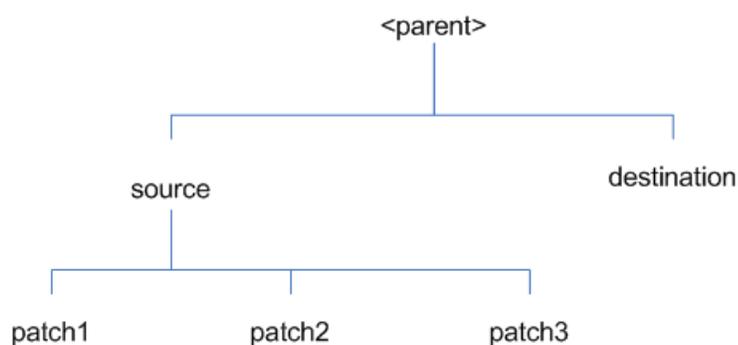
Source and Destination Directories

ORMerge uses source and destination areas in order to merge patch files. The source area is a single directory that contains the extracted patches to merge. The destination area is the location where the merged patch will be created. If a file exists in one or more source patches, only the highest revision will be copied to the merged patch.

The source and destination directories should exist under the same parent directory. That is, both the source and destination directories should be subdirectories of a single top-level directory.

The source directory must have all patches to be merged as immediate child directories. For example if three patches need to be merged the directory structure would look like this:

Source and Destination Directory Example



In the example above, the manifest.csv and patch_info.cfg files for each patch to be merged must exist in source/patch1, source/patch2, and source/patch3.

ORMerge Command-line Arguments

Argument	Required	Description
-s	Yes	Path to source directory containing patches to merge
-d	Yes	Path to destination directory that will contain merged patch
-name	No	The name to give the merged patch. If not specified, a name will be generated. When the merged patch is applied to a system, this name will appear in the Oracle Retail patch inventory.
-inplace	No	Used only when applying a patch to installation files prior to the first installation. See "Patching prior to the first install" in the Troubleshooting section later, for more information.

Running the ORMerge Utility

To merge patches, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Create a staging directory to contain the patches.

```
mkdir -p $RETAIL_HOME/stage/merge/src
```
5. Download the patches to the staging directory and unzip them so that each patch is in a separate subdirectory.
6. Review the README.txt included with each patch to identify additional manual steps that may be required. If manual steps are specified in any patch, execute them at the appropriate time when applying the merged patch.
7. Create a destination directory to contain the merged patches.

```
mkdir -p $RETAIL_HOME/stage/merge/dest
```
8. Execute ORMerge to merge the patches.

```
ormerge -s $RETAIL_HOME/stage/merge/src -d $RETAIL_HOME/stage/merge/dest -name merged_patch
```

The merged patch can now be applied as a single patch to the product installation using ORPatch.

Compiling Application Components

In some cases it may be desirable to recompile RMS Forms, RWMS Forms or RMS Batch outside of a product patch. The ORCompile utility is designed to make this easy and remove the need to manually execute 'make' or 'frmcmp' commands which can be error-prone. ORCompile leverages ORPatch functions to ensure that it compiles forms and batch exactly the same way as ORPatch. In addition ORCompile offers an option to compile invalid database objects using ORPatch logic.

ORCompile takes two required command line arguments each of which take an option. Arguments and options can be specified in upper or lower case.

ORCompile Command Line Arguments

Argument	Description
-a <app>	The application to compile.
-t <type>	The type of application objects to compile

ORCompile Argument Options

Application	Type	Description
RMS	BATCH	Compile RMS Batch programs
RMS	FORMS	Compile RMS Forms
RWMS	FORMS	Compile RWMS Forms
RMS	DB	Compile invalid database objects in the primary RMS schema
RMS	DB-ASYNC	Compile invalid database objects in the RMS_ASYNC_USER schema
ALLOC	DB-ALC	Compile invalid database objects in the Allocations user schema
ALLOC	DB-RMS	Compile invalid database objects in the RMS schema
REIM	DB	Compile invalid database objects in the RMS schema
RME	DB	Compile invalid database objects in the RME schema
ASO	DB	Compile invalid database objects in the ASO schema
RI	DB-DM	Compile invalid database objects in the RI DM schema
RI	DB-RIBATCH	Compile invalid database objects in the RI batch schema
RI	DB-RMSBATCH	Compile invalid database objects in the RI RMS batch schema
RI	DB-FEDM	Compile invalid database objects in the RI front-end schema
RDE	DB-DM	Compile invalid database objects in the RDE DM schema
RDE	DB-RDEBATCH	Compile invalid database objects in the RDE batch schema
RDE	DB-RMSBATCH	Compile invalid database objects in the RDE RMS batch schema

Note: Compiling RMS type DB, ReIM type DB, and Allocation type DB-RMS, are all identical as they attempt to compile all invalid objects residing in the RMS schema.

Running the ORCompile utility

To compile files, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```

- Execute orcompile to compile the desired type of files.

```
orcompile -a <app> -t <type>
```

ORCompile Examples

Compile RMS Batch.

```
orcompile -a RMS -t BATCH
```

Compile RWMS Forms.

```
orcompile -a RWMS -t FORMS
```

Compile invalid objects in the RA DM schema.

```
orcompile -a RI -t DB-DM
```

Compile invalid objects in the RMS owning schema.

```
orcompile -a RMS -t DB
```

Deploying Application Components

In some cases it may be desirable to redeploy Java applications outside of a product patch. For example, when troubleshooting a problem, or verifying the operation of the application with different WebLogic settings. Another situation might include wanting to deploy the application using the same settings, but without customizations to isolate behavior that could be related to customized functionality.

The ordeploy utility is designed to make this easy and remove the need to re-execute the entire product installer when no configuration needs to change. ORDeploy leverages Oracle Retail Patch Assistant functions to ensure that it deploys applications exactly the same way as ORPatch. In addition ORDeploy offers an option to include or not include custom Java files, to ease troubleshooting.

ORDeploy takes two required command line arguments each of which take an option. Arguments and options can be specified in upper or lower case.

ORDeploy Command Line Arguments

Argument	Description
-a <app>	The application to deploy.
-t <type>	The type of application objects to deploy

ORDeploy Argument Options

Application	Type	Description
ALLOC	JAVA	Deploy the Allocations Java application and Java batch files, including any custom Java files.
ALLOC	JAVANOCUSTOM	Deploy the Allocations Java application and Java batch files, NOT including any custom Java files.
REIM	JAVA	Deploy the REIM Java application and Java batch files, including any custom Java files.
REIM	JAVANOCUSTOM	Deploy the REIM Java application and Java batch files, NOT including any custom Java files.

Application	Type	Description
RESA	JAVA	Deploy the RESA Java application, including any custom Java files.
RESA	JAVANOCUSTOM	Deploy the RESA Java application, NOT including any custom Java files.
RPM	JAVA	Deploy the RPM Java application and Java batch files, including any custom Java files.
RPM	JAVANOCUSTOM	Deploy the RPM Java application and Java batch files, NOT including any custom Java files.

Running the ORDeploy utility

To deploy Java applications, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. Execute ORDeploy to deploy the desired Java application.

```
ordeploy -a <app> -t <type>
```

ORDeploy Examples

Deploy RPM.

```
ordeploy -a RPM -t JAVA
```

Deploy ReIM without including Java customizations.

```
ordeploy -a REIM -t JAVANOCUSTOM
```

Maintenance Considerations

The additional information stored within the RETAIL_HOME and within database schemas adds some considerations when performing maintenance on your environment.

Database Password Changes

Oracle wallets are used to protect the password credentials for connecting to database schemas. This includes all database schemas used during an install. If the password for any of these users is changed the wallet's entry must be updated.

The wallet location is configurable but by default is in the following locations:

Location	Installation Type
\$RETAIL_HOME/orphatch/rms_wallet	RMS Database RMS Batch
\$RETAIL_HOME/orphatch/rms_wallet_app	RMS Forms
\$RETAIL_HOME/orphatch/rwms_wallet	RWMS Database
\$RETAIL_HOME/orphatch/rwms_wallet_app	RWMS Forms
\$RETAIL_HOME/orphatch/oraso_wallet	ASO Database

Location	Installation Type
\$RETAIL_HOME/orpatch/orme_wallet	RME Database
\$RETAIL_HOME/orpatch/ra_wallet	RI (Previously RA) Database
\$RETAIL_HOME/orpatch/rde_wallet	RDE Database

The wallet alias for each schema will be <username>_<dbname>. Standard mkstore commands can be used to update the password.

For example:

```
mkstore -wrl $RETAIL_HOME/orpatch/rms_wallet -modifyCredential rms_rmsdb rms01
rmspassword
```

This command will update the password for the RMS01 user to 'rmspassword' in the alias 'rms_rmsdb'.

The Oracle wallets are required to be present when executing ORPatch. Removing them will prevent you from being able to run ORPatch successfully. In addition the Oracle wallet location is referenced in the RMS batch.profile, and in the default RMS and RWMS Forms URL configuration, so removing them will require reconfiguration of batch and forms. If batch and forms were reconfigured after installation to use other wallet files, it is possible to backup and remove the wallets, then restore them when running ORPatch.

WebLogic Password Changes

Java wallets are used to protect the password credentials used when deploying Java products. This includes the WebLogic administrator credentials, LDAP connection credentials, batch user credentials and any other credentials used during an install. If the password for any of these users is changed the wallet's entry must be updated, or the Java product installation can be run again.

The wallet location is in the following locations:

Location	Installation Type
\$RETAIL_HOME/orpatch/config/javapp_rpm	RPM Java
\$RETAIL_HOME/orpatch/config/javapp_reim	ReIM Java
\$RETAIL_HOME/orpatch/config/javapp_alloc	Allocation Java
\$RETAIL_HOME/orpatch/config/javapp_resa	RESA Java
\$RETAIL_HOME/orpatch/config/javaapp_rasrm	ORAAC (Previously RASRM) Java

The wallet aliases will be stored in the retail_installer partition. The names of the aliases will vary depending on what was entered during initial product installation.

The dump_credentials.sh script can be used to list the aliases in the wallet.

For example:

```
cd $RETAIL_HOME/orpatch/deploy/retail-public-security-api/bin
./dump_credentials.sh $RETAIL_HOME/orpatch/config/javapp_alloc
```

```
Application level key partition name:retail_installer
```

```
User Name Alias:dsallocAlias User Name:rms01app
```

```
User Name Alias:BATCH-ALIAS User Name:SYSTEM_ADMINISTRATOR
```

```
User Name Alias:wlsAlias User Name:weblogic
```

The easiest way to update the credential information is to re-run the Java product installer. If you need to manually update the password for a credential, the `save_credential.sh` script can be used.

For example:

```
cd $RETAIL_HOME/orpatch/deploy/retail-public-security-api/bin
./save_credential.sh -l $RETAIL_HOME/orpatch/config/javapp_alloc -p
retail_installer -a wlsAlias -u weblogic
```

This command will prompt for the new password twice and update the alias `wlsAlias`, username `weblogic` with the new password.

Infrastructure Directory Changes

The `RETAIL_HOME/orpatch/config/env_info.cfg` file contains the path to the database `ORACLE_HOME` on database or RMS Batch installations, to the WebLogic Forms and Reports `ORACLE_HOME` and `ORACLE_INSTANCE` on RMS or RWMS Forms installations, and to the `WEBLOGIC_DOMAIN_HOME`, `WL_HOME` and `MW_HOME` on Java product installations. If these paths change, the related configuration variables in the `env_info.cfg` file must be updated.

DBManifest Table

The table `dbmanifest` within Oracle Retail database schemas is used to track the database scripts which have been applied to the schema. It is critical not to drop or truncate this table. Without it, `ORPatch` will attempt to re-run scripts against the database which have already been applied which can destroy a working environment. Similarly, if copying a schema from one database to another database, ensure that the `dbmanifest` table is preserved during the copy.

RETAIL_HOME relationship to Database and Application Server

The `RETAIL_HOME` associated with an Oracle Retail product installation is critical due to the additional metadata and historical information contained within it. If a database or application installation is moved or copied, the `RETAIL_HOME` related to it should be copied or moved at the same time.

Jar Signing Configuration Maintenance

The RPM product installation includes an option to configure a code signing certificate so that jar files modified during installation or patching are automatically re-signed. This configuration is optional, but recommended. If it is configured, the code signing keystore is copied during installation to `$RETAIL_HOME/orpatch/config/jarsign/orpkeystore.jks`. The keystore password and private key password are stored in a Java wallet in the `$RETAIL_HOME/orpatch/config/jarsign` directory. The credentials are stored in a wallet partition called `orpatch`:

Alias	Username	Description
storepass	discard	Password for the keystore
keypass	discard	Password for the private key

The keystore file and passwords can be updated using the product installer. This is the recommended way to update the signing configuration.

If only the credentials need to be updated, the `sign_jar.sh` script can be used.

5. Log in as the UNIX user that owns the product installation.
6. Set the `RETAIL_HOME` environment variable to the top-level directory of your installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
7. Change directories to the location of `sign_jar.sh`

```
cd $RETAIL_HOME/orpatch/deploy/bin
```
8. Execute `sign_jar.sh`

```
sign_jar.sh changepwd
```
9. When prompted, enter the new keystore password
10. When prompted, enter the new private key password

Customization

Patching Considerations with Customized Files and Objects

In general, the additional capabilities provided by the ORPatch should make it easier to evaluate the potential impacts of patches to your customizations of Oracle Retail products. However, the additional metadata maintained by the Oracle Retail patching utilities does add some considerations when making customizations.

General Guidelines

It is always preferred to customize applications by extension rather than by direct modification. For example, adding new database objects and forms rather than modifying existing Oracle Retail objects and forms. You can also leverage built-in extension points such as User Defined Attributes, the Custom Flexible Attribute Solution, or seeded customization points in ADF Applications.

It is strongly discouraged to directly modify Oracle Retail database objects, especially tables, as your changes may be lost during patching or may conflict with future updates. When adding or modifying database objects, Oracle Retail recommends that all objects be added with scripts to ensure that they can be rebuilt if necessary after a patch.

Custom Database Objects

When you create new database objects, Oracle Retail recommends placing them in an Oracle database schema specifically for your customizations. You must use synonyms and grants to allow the Oracle Retail product schema owner and other users to access your objects, and use synonyms and grants to allow your customizations to access Oracle Retail objects. A separate schema will ensure that your customizations are segregated from base Oracle Retail code.

ORPatch expects that there will be no invalid objects in the database schemas it manages after a patch is applied. For this reason adding extra objects to the product schema could result in failures to apply patches as changes to base objects may cause custom objects to go invalid until they are updated. In this situation, manually update the custom objects so that they compile, and restart the patch.

Custom Forms

When creating new custom forms, Oracle Retail recommends placing them in a separate directory specifically for your customizations. This directory should be added to the FORMS_PATH of your RMS or RWMS Forms URL configuration to allow the forms to be found by the Forms Server. This will ensure that your customizations are segregated from base Oracle Retail code. If you choose to place customizations in the Forms bin directory, then your custom forms will need to be recopied each time Forms are fully recompiled.

ADF Application Customization

Oracle Retail ADF-based applications such as Allocation and ReSA can be customized using a process called 'seeded customization'. The customization process involves using JDeveloper in Customizer mode to create changes to product configurations, and then building a MAR archive containing the changes. The generated MAR is deployed to the MDS repository used by the application and applied to the application at runtime. These types of customizations are handled outside of ORPatch and are not reported during patch analysis or tracked by the custom file registration utility. More information can be found in the respective product customization guides.

Custom Compiled Java Code

When customizing Oracle Retail Java-based products such as RPM and ReIM via product source code, ORPatch supports automatically adding compiled customizations into the application ear file prior to deployment. This allows customizations to be applied to the application without directly modifying the base product ear, enabling customizations and defect hotfixes to co-exist when they do not change the same file or a dependent file. See the later “Custom Compiled Java Code” section for additional information and considerations.

Analyze Patches when Customizations are Present

Whenever you have customized a product by directly modifying Oracle Retail files or database objects, it is important to ensure you analyze each the files that will be updated by a patch before applying the patch. This will allow you to identify any customized files which may be overwritten by the patch and either merge your customization with the new version of the file, or re-apply the customization after applying the patch.

Manifest Updates

If you choose to customize Oracle Retail files directly, it is extremely important **not** to update the revision number contained in the env_manifest.csv. This could cause future updates to the file to be skipped, invalidating later patch applications as only a partial patch would be applied. The customized revision number for modified files will need to be tracked separately.

Registering Customized Files

The ORPatch contains utilities and functionality to allow tracking of files that have been customized through direct modification. This process is referred to as ‘registering’ a customized file. Registration only works for files which are shipped by Oracle Retail. It is not possible to register new files created in the environment as part of extensions or customizations.

When patches are analyzed with ORPatch, special reporting is provided if any registered files would be updated or deleted by the patch. Customized files impacted by the patch are listed at the end of the analysis report from ORPatch. The detail files generated during the analyze will contain a column called ‘customized’ which will have a Y for any files which were registered as customized. This allows easier identification of customizations which will be overwritten by a patch.

All files delivered by Oracle Retail are considered ‘base’ and so when they are applied to an environment any registrations of those files as customized will revert back to un-customized. **Each time a patch overwrites customized files, you must re-register the files as customized once you have applied customizations.**

To register customized files, use the \$RETAIL_HOME/orpatch/bin/orcustomreg script.

The orcustomreg script operates in one of two modes: registration and list.

- Registration mode registers or unregisters one or more files as customized.
- List mode lists all files in the environment that are registered as customized.

Command Line Arguments for Registration Mode

Argument	Description
-f <file>	Adds <file> to the list of files that will be registered. Can be specified more than once.

Argument	Description
-bulk <file>	Specifies a file to read, containing one filename per line. All filenames listed inside <file> will be registered.
-register	Files specified with -f or -bulk will be registered as 'customized'
-unregister	Files specified with -f or -bulk will be registered as 'base'

Notes:

- At least one of -f or -bulk is required.
 - If neither -register nor -unregister is specified, the default is '-register'.
 - File names specified with -f must either be fully-qualified or be relative to RETAIL_HOME. The same is true for filenames specified within a -bulk file.
-
-

Command Line arguments for list mode

Argument	Description
-list	List all files in the environment registered as customized

Running the orcustomreg Script

Perform the following procedure to run the orcustomreg script:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. Execute orcustomreg script to register the desired file(s).

```
orcustomreg -register -f <file>
```

Examples of using the orcustomreg Script

Register \$RETAIL_HOME/dbsql_rms/Cross_Pillar/control_scripts/source/oga.sql as customized.

```
orcustomreg -f dbsql_rms/Cross_Pillar/control_scripts/source/oga.sql
```

Unregister customizations for

```
$RETAIL_HOME/dbsql_rwms/Triggers/Source/TR_WAVE.trg
```

```
orcustomreg -unregister -f $RETAIL_HOME/dbsql_rwms/Triggers/Source/TR_WAVE.trg
```

Bulk register several files as customized.

```
echo "$RETAIL_HOME/oracle/proc/src/mrt.pc" > custom.txt
echo "$RETAIL_HOME/oracle/proc/src/saldly.pc" >> custom.txt
echo "$RETAIL_HOME/oracle/proc/src/ccprg.pc" >> custom.txt
orcustomreg -bulk custom.txt
```

List all files registered as customized.

```
orcustomreg -list
```

Custom Compiled Java Code

When customizing Oracle Retail Java-based products such as RPM and ReIM via product source code, ORPatch supports automatically adding compiled customizations into the application ear file prior to deployment. This allows customizations to be applied to the application without directly modifying the base product ear, enabling customizations and defect hotfixes to co-exist when they do not change the same file or a dependent file.

This functionality is enabled by creating a directory called `$RETAIL_HOME/javaapp_<app>/custom`, where `<app>` is the application the customizations apply to. Files stored within this directory will be combined with the base product ear files before the application is deployed to WebLogic. ORPatch will attempt to consider customizations stored within the 'custom' directory during patch analysis by triggering more detailed ear file change analysis to assist with identifying which customizations might be impacted by changes in the patches.

Note: It is not possible, nor necessary, to register compiled Java customizations with the `orcustomreg` tool.

As with other customization techniques for other technologies, Oracle Retail recommends making Java customizations in new files as much as possible, versus overwriting base product or configuration files. In the past it was necessary to build complete replacement product ear files, but this method of customization is no longer required nor recommended. Replacement ear and jar files will not contain the `META-INF/env_manifest.csv` files which are required in order to be able to apply incremental patches. Instead, compile the specific Java classes being customized and place them along with any custom configuration files in `$RETAIL_HOME/javaapp_<app>/custom`.

Building Deployable ear files

When constructing the product ear file to deploy to WebLogic, ORPatch applies changes to the ear file in a specific order, with files from later steps overwriting files in earlier steps. The resulting ear is stored in `$RETAIL_HOME/javaapp_<app>/deploy`, and then deployed to WebLogic.

Sequence for ORPatch Java Product ear file updates

Order	File Type	Location
1	Base product ear	<code>\$RETAIL_HOME/javaapp_<app>/base</code>
2	Updated configuration files	<code>\$RETAIL_HOME/javaapp_<app>/config</code>
3	Oracle Retail-supplied hotfixes	<code>\$RETAIL_HOME/javaapp_<app>/internal</code>
4	Compiled customizations	<code>\$RETAIL_HOME/javaapp_<app>/custom</code>

Merging Custom Files

When merging files from the custom directory with the product ear, ORPatch uses the directory path of the files within custom to calculate where the file should be stored within the ear. This allows arbitrary nesting of files, even when placing files within jars stored in jars, stored within the ear. The following examples below use RPM, but apply to adding compiled customizations to any Java-based product.

Custom directory location and product ear location Examples

File path within javaapp_<app>/custom/	Final Ear File Location
rpm.ear/company/ui/MyCustom.class	In rpm.ear: /company/ui/MyCustom.class
rpm.ear/rpm.jar/company/bc/MyCustom2.class	In rpm.ear: In rpm.jar: /company/bc/MyCustom2.class
rpm.ear/lib/ourcustomlibs.jar	In rpm.ear /lib/ourcustomlibs.jar
rpm.ear/WebLaunchServlet.war/lib/ rpm.jar/company/bc/MyCustom2.class	In rpm.ear: In WebLaunchServlet.war: In lib/rpm.jar: /company/bc/MyCustom2.class

Analyzing patches when customizations are present

When analyzing a patch which contains a base product ear and the custom directory contains files, ORPatch will automatically trigger a more detailed analysis of the changes coming in a patch. This includes calculating what files inside the product ear have been added, removed or updated and which files appear to be customized based on the contents of the 'custom' directory. The detailed results of the ear file comparison during patch analysis will be saved in javaapp_<app>_archive_compare_details.csv. Any custom files which appeared to be impacted by the patch are saved in javapp_<app>_archive_custom_impacts.csv. Both files will be in the \$RETAIL_HOME/orpatch/logs/detail_logs/analyze/details directory.

Note: This detailed analysis is not available when analyzing individual hotfixes, so special care must be taken when applying hotfixes to a customized product installation, to ensure there are no conflicts between customizations and hotfix changes.

Customizations and cumulative patches

By default, when applying a cumulative patch, ORPatch will not include customizations in the deployed product ear, even if they are present in the appropriate directory. This allows verification that the application is functioning properly using base code, before applying customizations. After verifying the initial deployment, use ORDeploy with the "-t JAVA" option to construct and deploy the product ear including customizations.

If customizations need to be removed outside of a patch, use ORDeploy with the "-t JAVANOCUSTOM" option to create and deploy an ear containing only Oracle Retail code. To force ORPatch to include customizations in the deployed ear even when applying a cumulative patch, set JAVAAPP_<app>_INCLUDE_CUSTOM=Y in the \$RETAIL_HOME/orpatch/config/env_info.cfg file.

Changing configuration files

It is possible to directly change product configuration files in \$RETAIL_HOME/javaapp_<app>/config. These updates can be deployed to the environment using the ORDeploy utility. However, the 'config' directory is completely

recreated each time the product installer is used. This means that modifications will be lost and must be manually reapplied after each installer run. It is recommended to make configuration changes via the installer where possible, and retain the `ant.install.properties` file for use in later installer sessions.

Extending Oracle Retail Patch Assistant with Custom Hooks

The default ORPatch actions and processing logic is sufficient to install and patch the base Oracle Retail product code. However there may be situations where custom processing is desired during patching activities such as executing a shell script prior to the start of patching, or running a SQL script at the end of the patch.

ORPatch supports extensions in the form of custom hooks. These hooks allow external scripts to be run at specific points during ORPatch processing.

ORPatch Processing

Action

ORPatch supports a variety of 'actions' which define the steps necessary to apply updates to a particular area of the Oracle Retail application. Each action is generally specific to updates to a single technology or logical component of the environment. For example, one action might handle making updates to the RMS database schema, while a separate action is responsible for compiling RWMS forms, and a different action deploys the RPM Java application. These actions are enabled and disabled within the environment configuration file, allowing ORPatch to determine what types of changes to apply to each product installation.

ORPatch Actions

Order	Action Name	Description
1	DBSQL_RMS	Loads RMS and RPM database objects into the primary RMS schema
2	DBSQL_RMSASYNC	Loads database objects into the RMS_ASYNC_USER schema
3	DBSQL_RAF	Loads Retail Application Framework database objects into the RMS schema
4	DBSQL_REIM	Loads ReIM database objects into the RMS schema
5	DBSQL_ALCRMS	Loads Allocation database objects into the RMS schema
6	DBSQL_ALLOC	Loads Allocation database objects into the Allocation user schema
7	DBSQL_RMSDEMO	Used to create demo data in the RMS schema if demo data was selected during initial installation
8	DBSQL_RMSDAS	Loads database objects into the RMS Data Access Schema
9	RMSBATCH	Compiles RMS Batch
10	ORAFORMS_RMS	Compiles RMS Forms, copies RMS reports to \$RETAIL_HOME
11	RMSRETLSCRIPTS	Copies Oracle Retail Extract and Load scripts for RMS
12	RMSDCSCRIPTS	Copies Oracle Retail Merchandising System data conversion scripts

Order	Action Name	Description
13	DBSQL_RWMS	Loads database objects into the primary RWMS schema
14	DBSQL_RWMSADF	Loads database objects into the RWMS ADF user schema
15	DBSQL_RWMSUSER	Loads database objects into the RWMS user schema
16	ORAFORMS_RWMS	Compiles RWMS Forms, copies RWMS batch scripts and reports to \$RETAIL_HOME
17	JAVAAPP_RPM	Deploys the RPM Java application and batch scripts
18	JAVAAPP_REIM	Deploys the REIM Java application and batch scripts
19	JAVAAPP_ALLOC	Deploys the Allocation Java application and batch scripts
20	JAVAAPP_RESA	Deploys the ReSA Java application
21	JAVAAPP_RASRM	Deploys the ORAAC (previously called RASRM) Java application
22	DBSQL_RARMSBATCH	Loads database objects into the RMS Batch schema for RI (previously called RA)
23	DBSQL_RADM	Loads database objects into the RI (previously called RA) Data Mart schema
24	DBSQL_RAFEDM	Loads database objects into the RI (previously called RA) Front-end schema
25	DBSQL_RABATCH	Loads database objects into the RI (previously called RA) Batch schema
26	DBSQL_RDERMSBATCH	Loads database objects into the RMS Batch schema for RDE
27	DBSQL_RDEDM	Loads database objects into the RDE Data Mart schema
28	DBSQL_RDEBATCH	Loads database objects into the RDE Batch schema
29	DBSQL_RASECORE	Loads core database objects into the ORASE schema
30	DBSQL_RASEASO	Loads ASO database objects into the ORASE schema
31	DBSQL_RASECDT	Loads CDT database objects into the ORASE schema
32	DBSQL_RASECIS	Loads CIS database objects into the ORASE schema
33	DBSQL_RASEDT	Loads DT database objects into the ORASE schema
34	DBSQL_RASEMBA	Loads MBA database objects into the ORASE schema
35	RASECOREBATCH	Copies ORASE core batch scripts and libraries
36	RASEASOBATCH	Copies ORASE ASO batch scripts and libraries
37	RASECDTBATCH	Copies ORASE CDT batch scripts and libraries
38	RASECISBATCH	Copies ORASE CIS batch scripts and libraries
39	RASEDTBATCH	Copies ORASE DT batch scripts and libraries
40	RASEMBABATCH	Copies ORASE MBA batch scripts and libraries

Phase

ORPatch processes patches in phases. Each action relevant to a patch and host is provided an opportunity to process the patch for each phase. The standard phases which allow hooks are:

Restart Phase Number	Phase Name	Description
N/A	PRECHECK	Actions verify that their configuration appears complete and correct. This phase and the associated hooks will be run every time orpatch is executed, even if processing will be restarted in a later phase.
10	PREACTION	Actions do processing prior to when files are copied to the environment. Files are deleted during this phase.
20	COPYPATCH	Actions copy files included in a patch into the destination environment and the environment manifest is updated.
30	PATCHACTION	Actions take the more detailed steps necessary to apply the new files to the environment. For database actions in particular, this is the phase when new and updated sql files are loaded into the database.
40	POSTACTION	Actions do processing after files have been copied and PatchActions are completed. The Forms actions, for example, use this phase to compile the forms files as this must happen after database packages are loaded.
50	CLEANUP	Actions do any additional processing. Currently no actions implement activities in this phase.

Configuring Custom Hooks

Custom hooks are configured in a configuration file `RETAIL_HOME/orpatch/config/custom_hooks.cfg`. The configuration file is a simple text file where blank lines and lines starting with `#` are ignored and all other lines should define a custom hook.

To define a custom hook, a line is added to the file in the form:

```
<hook name>=<fully qualified script>
```

The hook name must be in upper case and is in the form:

```
<action name>_<phase name>_<sequence>
```

The action name is any action name understood by ORPatch. The phase name is one of the five phase names from the table above. The sequence is either 'START' or 'END'. Hooks defined with a sequence of 'START' are run before the action's phase is invoked. Hooks defined with a sequence of 'END' are run after the action's phase is invoked.

Multiple scripts can be associated with a single hook by separating the script names with a comma. If a hook name appears in the configuration file multiple times only the last entry will be used.

The script defined as a custom hook must be an executable shell script that does not take any arguments or inputs. The only environment variable that is guaranteed to be passed to the custom hook is `RETAIL_HOME`. The script must return 0 on success and non-zero on failure.

If an action is a DBSQL action (i.e. has a name like `DBSQL_`), the custom hook can optionally be a `.sql` file. In this case the SQL script will be run against the database schema that the DBSQL action normally executes against. The SQL script must not

generate any ORA- or SP2- errors on success. In order to be treated as a database script, the extension of the file defined as the custom hook must be .sql in lower-case. Any other extension will be treated as if it is a shell script. If you have database scripts with different extensions, they must be renamed or wrapped in a .sql script.

When using the PRECHECK phase and START sequence, please note that the custom hook will be executed prior to any verification of the configuration. Invalid configuration, such as invalid database username/password or a non-existent ORACLE_HOME, may cause the custom hook to fail depending on the actions it tries to take. However in these cases, the normal orpatch PRECHECK activities would likely have failed as well. All that is lost is the additional context that orpatch would have provided about what was incorrect about the configuration.

Restarting with Custom Hooks

If a custom hook fails, for example a shell script hook returns non-zero or a sql script generates an ORA- error in its output, the custom hook will be treated as failing. A failing custom hook causes ORPatch to immediately stop the patching session.

When ORPatch is restarted it always restarts with the same phase and action, including any START sequence custom hooks. If the START sequence custom hook fails, the action's phase is never executed. With an END sequence custom hook, the action's phase is re-executed when ORPatch is restarted and then the custom hook is re-executed.

When an action's phase is costly, for example the DBSQL_RMS action which does a lot of work, this can mean a lot of duplicate processing.

For this reason it is preferred to use START sequence custom hooks whenever possible. If necessary, use a START sequence hook on a later phase or a later action, rather than an END sequence custom hook.

Patch-level Custom Hooks

In addition to action-specific hooks, there are two patch-level hook points available. These hooks allow scripts to be run before any patching activities start and after all patching activities are completed. The hooks are defined in the same configuration file, with a special hook name.

To run a script before patching, define:

```
ORPATCH_PATCH_START=<fully qualified script>
```

To run a script after patching, define:

```
ORPATCH_PATCH_END=<fully qualified script>
```

These hooks only support executing shell scripts, database scripts must be wrapped in a shell script. It is also important to note that these hooks are run on every execution of ORPatch to apply a patch, even when restarting a patch application. If the START sequence patch-level hook returns a failure, patching is aborted. If the END sequence patch-level hook returns a failure, it is logged but ignored as all patching activities have already completed.

Please note that the ORPATCH_PATCH_START hook is executed prior to any verification of the configuration. Invalid configuration may cause the custom hook to fail depending on the actions it tries to take. However in these cases, the normal ORPatch activities would likely fail as well.

Example Custom Hook Definitions

A shell script that is executed prior to the Pre-Action phase of RMS Batch:

```
RMSBATCH_PREACTION_START=/u00/oretail/prepare_custom_header.sh
```

A shell script that is executed after RETL script files are copied into the RETAIL_HOME:

```
RETLSCRIPTS_COPYPATCH_END=/u00/oretail/copy_custom_files.sh
```

A SQL script that is executed against the RWMS owning schema at the start of the Clean-up Phase:

```
DBSQL_RWMS_CLEANUP_START=/dba/sql/recompile_synonyms.sql
```

Troubleshooting Patching

There is not a general method for determining the cause of a patching failure. It is important to ensure that patches are thoroughly tested in a test or staging system several times prior to attempting to apply the patch to a production system, particularly if the patch is a large cumulative patch. After the test application is successful, apply the patch to the production system.

ORPatch Log Files

ORPatch records extensive information about the activities during a patch to the log files in `RETAIL_HOME/orpatch/logs`. This includes a summary of the actions that are planned for a patch, information about all files that were updated by the patch, and detailed information about subsequent processing of those files. The ORPatch log files also contain timestamps to assist in correlating log entries with other logs.

Even more detailed logs are available in `RETAIL_HOME/orpatch/logs/detail_logs` for some activities such as forms compilation, invalid database object errors, and output from custom hooks. If the standard ORPatch log information is not sufficient, it might be helpful to check the detailed log if it exists.

Restarting ORPatch

The restart mechanism in ORPatch is designed to be safe in nearly any situation. In some cases to ensure this, a portion of work may be redone. If the failure was caused by an intermittent issue that has been resolved, restarting ORPatch may be sufficient to allow the patch to proceed.

Manual DBManifest Updates

A possible cause for database change script failures is that a database change was already made manually to the database. In this event, you may need to update the dbmanifest table to record that a specific script does not need to be run. Before doing this, it is extremely important to ensure that all statements contained in the script have been completed.

Use the `$RETAIL_HOME/orpatch/bin/ordbmreg` script to register database scripts in the dbmanifest table.

Command Line Arguments for ordbmreg

Argument	Description
<code>-f <file></code>	Adds <file> to the list of files that will be registered. Can be specified more than once.
<code>-bulk <file></code>	Specifies a file to read, containing one filename per line. All filenames listed inside <file> will be registered.
<code>-register</code>	Files specified with <code>-f</code> or <code>-bulk</code> will be registered in the dbmanifest table
<code>-unregister</code>	Files specified with <code>-f</code> or <code>-bulk</code> will be removed from the dbmanifest table

Notes:

- At least one of -f or -bulk is required.
 - If neither -register nor -unregister is specified, the default is '-register'.
 - File names specified with -f must either be fully-qualified or be relative to RETAIL_HOME. The same is true for filenames specified within a -bulk file.
 - Registering a file in the dbmanifest table will cause it to be completely skipped. Before doing so, ensure that all commands contained in it have been completed.
 - Removing a file from the dbmanifest table will cause it to be run again. This will fail if the commands in the script cannot be re-run. For example if they create a table that already exists.
-
-

Running the ordbmreg Script

Perform the following procedure to run the ordbmreg script:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. Execute ordbmreg script to register the desired file(s).

```
ordbmreg -register -f <file>
```

Examples of using the ordbmreg Script

Register

\$RETAIL_HOME/dbsql_rms/Cross_Pillar/db_change_scripts/source/000593_system_options.sql with the dbmanifest table.

```
ordbmreg -f
dbsql_rms/Cross_Pillar/db_change_scripts/source/000593_system_options.sql
```

Remove the dbmanifest row for

\$RETAIL_HOME/dbsql_radm/ra_db/radm/database_change_scripts/000035_s12733240_w_party_per_d.sql.

```
ordbmreg -unregister -f
$RETAIL_HOME/dbsql_radm/ra_db/radm/database_change_scripts/000035_s12733240_w_party_per_d.sql
```

Bulk register several files in the dbmanifest table.

```
echo "$RETAIL_HOME/dbsql_rwms/DBC/Source/000294_container.sql" > dbcs.txt
echo "$RETAIL_HOME/dbsql_rwms/DBC/Source/000457_drop_object.sql" >> dbcs.txt
ordbmreg -bulk dbcs.txt
```

Restarting after registration

Once the row has been added to the dbmanifest table, restart ORPatch and the script will be skipped. If the file is not skipped there are several possibilities:

- The script registered is not the failing script.
- The file type is not a type that is filtered by the dbmanifest. The only file types that skip files listed in the dbmanifest are:
 - Initial install DDL Files
 - Installation scripts that cannot be rerun
 - Database Change Scripts

Manual Restart State File Updates

Oracle Retail strongly discourages manually updating the ORPatch restart state files. Updating the file improperly could cause necessary steps in the patching process to be skipped or patches to be incorrectly recorded as applied.

DISPLAY Settings When Compiling Forms

When compiling RMS or RWMS forms, it is necessary to have a valid X-Windows Display. ORPatch allows this setting to come from one of two places:

- DISPLAY environment variable set before executing ORPatch
- or
- DISPLAY setting in RETAIL_HOME/orpatch/config/env_info.cfg

The DISPLAY variable in the environment overrides the env_info.cfg, if both are set. The destination X-Windows display must be accessible to the user running ORPatch, and for best compilation performance it should be on the network 'close' to the server where RMS Forms are installed and compiled. Using a local display or VNC display is preferred. Compiling forms across a Wide-Area Network will greatly increase the time required to apply patches to environments.

JAVA_HOME Setting

When working with Java application jar, ear or war files, it is necessary to have a valid JAVA_HOME setting. ORPatch allows this setting to come from one of two places:

- JAVA_HOME environment variable set before executing ORPatch
- or
- JAVA_HOME setting in RETAIL_HOME/orpatch/config/env_info.cfg

The JAVA_HOME variable in the environment overrides the env_info.cfg, if both are set. The specified Java home location must be accessible to the user running ORPatch and be a full Java Development Kit (JDK) installation. The JAVA_HOME must contain the jar utility and if automatic Jar file signing is configured, must also contain the keytool and jarsigner utilities.

Patching Prior to First Install

In some situations, it may be necessary to apply a patch to product installation files before the initial install. For example, if there is a defect with a script that would be run during the install and prevent proper installation. In this rare situation, it may be necessary to apply a patch to the installation files prior to starting installation.

Note: These steps should only be undertaken at the direction of Oracle Support.

Perform the following steps to patch installation files prior to starting an installation. The steps assume an RMS installation, but apply to any product supported by ORPatch:

1. Unzip the installation files to a staging area.

Note: The following steps assume the files are in
/media/oretail15.0

2. Locate the patch_info.cfg within the product media. The directory it resides in will be used for later steps.

```
find /media/oretail15.0/rms/installer -name patch_info.cfg
```

Output Example:

```
/media/oretail15.0/rms/installer/mom15/patch_info.cfg
```

3. Get the PATCH_NAME for the standard product installation. The patch name to use in subsequent steps will be the portion following the "=" sign.

```
grep "PATCH_NAME=" /media/oretail15.0/rms/installer/mom15/patch_info.cfg
```

Output Example:

```
PATCH_NAME=MOM_15_0_0_0
```

4. Create a directory that will contain the patch that must be applied, next to the directory with the product installation files.

Note: The following steps assume this directory is in
/media/patch.

5. Unzip the patch into the directory created in step 2.

Note: This should place the patch contents in
/media/patch/<patch num>.

6. Export RETAIL_HOME to point within the installation staging area.

```
export RETAIL_HOME=/media/oretail15.0/rms/installer/mom15/Build
```

7. Create a logs directory within the installation staging area

```
mkdir $RETAIL_HOME/orpatch/logs
```

8. Ensure the ORMerge shell script is executable.

```
chmod u+x $RETAIL_HOME/orpatch/bin/ormerge
```

9. Run ORMerge to apply the patch to the installation media, using a -name argument that is the same as what was found in step 3.

```
$RETAIL_HOME/orpatch/bin/ormerge -s /media/patch -d  
/media/oretail15.0/rms/installer/mom15 -name MOM_15_0_0_0 -inplace
```

Note: The -inplace argument is critical to ensure that the
patching replaces files in the mom15 directory.

10. Unset the RETAIL_HOME environment variable.

```
unset RETAIL_HOME
```

At this point, the installation files will have been updated with the newer versions of files contained within the patch. Log files for the merge will be in
/media/oretail15.0/rms/installer/mom15/Build/orpatch/logs.

Providing Metadata to Oracle Support

In some situations, it may be necessary to provide details of the metadata from an environment to Oracle support in order to assist with investigating a patching or application problem. ORPatch provides built-in functionality through the 'exportmetadata' action to extract and consolidate metadata information for uploading to

Oracle Support or for external analysis. For more information, see the ORPatch 'Exporting Environment Metadata' section.

Appendix: Oracle 12cR1 Database Parameter File

```
#####
# Copyright (c) 2015 by Oracle Corporation
# Oracle 12.1.0.x Parameter file
# NOTES: Before using this script:
#       1. Change <datafile_path>, <admin_path>, <utl_file_path>, <diag_path>
#       and <hostname>
#           values as appropriate.
#       2. Replace the word SID with the database name.
#       3. Size parameters as necessary for development, test, and production
#       environments.
# -----
*.audit_file_dest=full_path_of_audit_dir
*.compatible='12.1.0.2'
*.control_files='full_path_of_controlfile_1','full_path_of_controlfile_2'
#####
# Memory Settings:
# xxxM = Some reasonable starting value for your environment
#####
*.db_block_size=xxxM
*.db_cache_size=xxxM
*.java_pool_size=xxxM
*.memory_target=xxxM
*.pga_aggregate_target=xxxM
*.shared_pool_size=xxxM
*.streams_pool_size=xxxM

#####

*.db_block_size=8192
*.db_domain='' # specify if multiple domains
*.db_name='dbName'
*.diagnostic_dest='full_path_of_diag_dir'
*.enable_pluggable_database=true|false
*.fast_start_mttr_target=xxx # mean time to recover in second.
*.nls_calendar='GREGORIAN'
*.nls_date_format='DD-MON-RR'
*.nls_language='AMERICAN'
*.nls_numeric_characters=',.'
*.nls_sort=BINARY
*.open_cursors=900
*.plsql_optimize_level=2
*.processes=2000
*.query_rewrite_enabled='true'
*.remote_dependencies_mode='SIGNATURE'
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_management=AUTO #Default: Manual
*.undo_retention=xxxx #Default: 900
*.undo_tablespace='specify undo_tbs'
```

Appendix: Configure Listener for External Procedures

Note: This example illustrates the listener configuration required for external procedures. It does not include environment specific settings that may be needed. Consult Oracle Net Services guides for additional information.

```
#####
# File: listener.ora
# Desc: Oracle Net8 listener file.
# Notes: Modify <hostname>
#####

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (PROTOCOL_STACK =
        (PRESENTATION = TTC)
        (SESSION = NS))
      (ADDRESS =
        (PROTOCOL = tcp)
        (HOST = <hostname>)
        (PORT = 1521))
      (ADDRESS =
        (PROTOCOL = IPC)
        (KEY = extproc_key))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (PROGRAM = extproc)
      (SID_NAME = extproc_agent)
      (ENVS='EXTPROC_DLLS=ANY')
    )
  )
)
```

Note: This example illustrates the configuration of net services names required for external procedures. It does not include environment specific settings that may be needed. Consult Oracle Net Services guides for additional information

```
#####
# File: tnsnames.ora
# Desc: Net Services configuration file.
# Note: Change these values: <service_name>, <oracle_sid>, <hostname>,
#       <global_name>
#####

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

EXTPROC_CONNECTION_DATA.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

< Connect_string> =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = <hostname>)(Port = 1521)))
    (CONNECT_DATA = (Service_Name = <Service_Name>) (GLOBAL_NAME =
<global_name>)))

<Connect_String>.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = <hostname>)(Port = 1521)))
    (CONNECT_DATA = (Service_Name = <Service_Name> ) (GLOBAL_NAME =
<global_name>)))

Example:
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

EXTPROC_CONNECTION_DATA.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

prod_dbl =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = server_01)(Port = 1521)))
    (CONNECT_DATA = (Service_Name = prod_dbl) (GLOBAL_NAME = prod_dbl.world)))

prod_dbl.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = server_01)(Port = 1521)))
    (CONNECT_DATA = (Service_Name = prod_dbl) (GLOBAL_NAME = prod_dbl.world)))
```

Appendix: Tablespace Creation

Non-Encrypted Tablespace Creation

Standard RMS tablespaces are created using the `create_rms_tablespaces.sql` script located in `STAGING_DIR/rms/installer/create_db`.

1. Modify `STAGING_DIR/rms/installer/create_db/create_rms_tablespaces.sql`. The table below shows the default initial sizes.

TABLESPACE_NAME	Size
ENCRYPTED_RETAIL_INDEX	12G
ENCRYPTED_RETAIL_DATA	10G
RETAIL_INDEX	10G
RETAIL_DATA	8G
LOB_DATA	2G
USERS	2G

2. Once this script has been modified, execute it in SQL*Plus as sys.
3. Review `create_rms_tablespaces.log` for errors and correct as needed.

Encrypted Tablespace Creation

If you do not have an Advanced Security Option license, create the `encrypted_retail_data` and `encrypted_retail_index` tablespaces as normal tablespaces.

1. Modify `STAGING_DIR/rms/installer/create_db/create_encrypted_tablespaces_no_TDE.sql`
2. Run the script using SQL*Plus as sys
3. Review `Create_encrypted_retail_tablespaces_no_TDE.log` for errors and correct as needed

With an Advanced Security license, tablespaces can be created in an encrypted format. The steps are:

Configure a Wallet

1. Create a `sqlnet.ora` in `$TNS_ADMIN` directory of the database server similar to the below entry:

```
ENCRYPTION_WALLET_LOCATION =
  (SOURCE = (METHOD = FILE)
   (METHOD_DATA =
    (DIRECTORY = /u00/oracle/admin/ORACLE_SID/wallet)))
```

2. Create the wallet directory:


```
mkdir -p /u00/oracle/admin/<ORACLE_SID>/wallet
```

3. As a user with the 'alter system' privilege, create the wallet as follows:

Non-container databases:

- a. ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
'/u00/oracle/admin/dbName/wallet' IDENTIFIED BY "pwd#";
- b. KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "pwd#";
- c. KEY MANAGEMENT SET KEY IDENTIFIED BY "pwd#" WITH BACKUP;
- d. ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE
FROM KEYSTORE '/u00/oracle/admin/dbName/wallet' identified by pwd#;
- a. Container databases:
- b. ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
'/u00/oracle/admin/dbName/wallet' IDENTIFIED BY "pwd#";
- c. ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE
FROM KEYSTORE '/u00/oracle/admin/dbName/wallet' identified by "pwd#";
- d. ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
"pwd#" Container=ALL;
- e. ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "pwd#" WITH
BACKUP USING 'TDE_ENCRYPTION' Container=all;

4. Confirm if the wallet is created and open (the TDE master encryption key has been created and inserted automatically):

```
SQL>
select substr(wrl_type, 1, 10) wrl_type, substr(wrl_parameter, 1, 45) param,
substr(status, 1, 10) status, substr(wallet_type, 1, 15) w_type
from v$encryption_wallet;
```

WRL_TYPE	PARAM	STATUS	W_TYPE
FILE	/u00/oracle/admin/ORACLE_SID/wallet	OPEN	AUTOLOGIN

An auto-open wallet is created. You are ready to create the encrypted tablespaces as shown in the following section.

Encryption at Tablespace Level

Once the wallet is configured, determine an encryption algorithm to be used for the encrypted tablespace and then create them. The sample scripts use the default algorithm AES128:

1. Modify STAGING_DIR/rms/installer/create_db/create_encrypted_tablespaces_TDE.sql.
2. Run the script using SQL*Plus as sys.
3. Review Create_encrypted_retail_tablespaces_TDE.log for errors and correct as needed.

Once the tablespaces have been created, the RMS schema installation can be run.

Note: After encryption at the tablespace level, it is absolutely crucial to backup the contents in the wallet directory; otherwise, if they are lost you will not be able to access the tablespaces.

Appendix: RMS RETL Instructions

This appendix summarizes the RETL program features utilized in the RMS Extractions (RMS ETL). More information about the RETL tool is available in the *Oracle Retail Extract, Transform, and Load Programmer's Guide*. More information about RMS ETL is available in the *Oracle Retail Merchandising System Operations Guide*.

Configuration: RETL

Before trying to configure and run RMS ETL, install RETL (version 13.2.8.0.1), which is required to run RMS ETL. For installation instructions, see Chapter 2 of the *Oracle Retail Extract, Transform, and Load Programmer's Guide*. Run the `verify_retl` script (included as part of the RETL installation) to ensure that RETL is working properly before proceeding. RETL 13.2.8.0.1 creates a wallet under `$RFX_HOME/etc/security`, with the following files:

- `cwallet.sso`
- `jazn-data.xml`
- `jps-config.xml`
- `README.txt`

To set up RETL wallets, complete the following steps:

1. Set the following environment variables:
 - `ORACLE_SID=retaildb`
 - `RFX_HOME=/u00/rfx/rfx-14.1`
 - `RFX_TMP=/u00/rfx/rfx-14.1/tmp`
 - `JAVA_HOME=/usr/java/jdk1.8.64bit`
 - `LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH`
 - `PATH=$RFX_HOME/bin:$JAVA_HOME/bin:$PATH`
2. Change directory to `$RFX_HOME/bin`.
3. Run `setup-security-credential.sh` as follows.
 - a. Enter 1 to add a new database credential.
 - b. Enter the dbuseralias (for example, `retl_java_rms01user`).
 - c. Enter the database user name (for example, `rms01user`).
 - d. Enter the database password.
 - e. Re-enter the database password.
 - f. Enter D to exit the setup script.
4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

For example, to configure RETLforRPAS, modify the following entries in `$RETAIL_HOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env`.

- The `RETL_WALLET_ALIAS` should point to the Java wallet entry:

```
export RETL_WALLET_ALIAS="retl_java_rms01"
```

- The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry in \$RETAIL_HOME/orpatch/rms_wallet:

```
export ORACLE_WALLET_ALIAS="retaildb rms01"
```

Note: See the section, [Setting Up Wallets for Database User Accounts](#).

- The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:

```
export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"
```

Note: When connecting to a pluggable database, the JDBCConn property in the .env file needs to be properly set. This requires <port>/<service_name> instead of <port>:<sid>. Below is an example:

```
export JDBCConn="<<PROPERTY name=\`jdbcconnectionstring\`
value=\`jdbc:oracle:thin:@msp28165.us.oracle.com:1521/retaildb\`"/>"
```

5. To change a password later, run setup-security-credential.sh as follows.
 - a. Enter 2 to update a database credential.
 - b. Select the credential to update.
 - c. Enter the database user to update or change.
 - d. Enter the password of the database user.
6. Re-enter the password.
7. Note the following, which is how the setup-security-credential.sh script looks as it runs.

```
/u00/rfx/rfx-14.1/bin> ./setup-security-credential.sh
```

```
=====
RETL Database Credentials Configuration Utility.
=====
```

Please select one of the below option:

- 1) Add a new database credentials
- 2) Modify/Update existing database credentials
- 3) Delete existing database credentials

([1], [2], [3], [D]one): 1

Please enter the dbuseralias (This has to be unique for each database):
 <oracle_sid>_<database user name>, i.e., retl_java_rms01

Please enter the database username: <database user name>, i.e., rms01

Please enter the database password (password text will not be displayed as it is entered) :

Verify database password :

Created the credentials for dbuseralias "retl_java_rms01" successfully

Please select one of the below option:

- 1) Add a new database credentials
- 2) Modify/Update existing database credentials
- 3) Delete existing database credentials

([1], [2], [3], [D]one): /u00/rfx/rfx-14.1/bin>

To run the RETL wallet, the /RETLforRPAS/rfx/etc/rmse_rpas_config.env file needs to be edited with the following entries included:

```
##The folloiwng setting is for dbuseralias being replaced for connectstring
and dbuserid
export RETL_WALLET_ALIASES=" retl_java_rms01"
```

Note: The following is an example of how to run a sample RETL script.

- To run a RETL script, set up your environment with the following run-time variables.

```
export RFX_HOME = i.e., /u00/rfx/rfx-14.1
export RFX_TMP = i.e., /u00/rfx/rfx-14.1/tmp
export TNS_ADMIN = i.e., $RETAIL_HOME/orpatch/rms_wallet
export
ALCHOME = i.e.,
/u00/webadmin/product/10.3.6/WLS/user_projects/domains/APPDomain/alloc14/rpas-
interfaces
export
RETAIL_HOME = i.e.,
/u00/webadmin/product/10.3.6/WLS/user_projects/domains/APPDomain/alloc14/rpas-
interfaces
export ORACLE_HOME = i.e., /u00/oracle/product/12.0.1
export JAVA_HOME = i.e., /usr/jdk1.7.0_45.64bit
export PATH = $ORACLE_HOME/bin:$JAVA_HOME/bin:$RFX_HOME/bin:$PATH
export
LD_LIBRARY_PATH = i.e.,
$RFX_HOME/lib:$ORACLE_HOME/lib:$RETAIL_HOME/oracle/lib/bin:/lib:/usr/lib:/usr/
dt/lib:/usr/openwin/lib
export TEMP_DIR = i.e., /home/alcbatch/rpas/tmp
export PATH = i.e., ${ORACLE_HOME}/bin:${JAVA_HOME}/bin:${PATH}
```

Go to \$ALCHOME/log and \$ALCHOME/error and delete all existing files.

Go to \$ALCHOME/rfx/src and run the following command:

```
>alcl_plan.ksh plan_01.dat
```

To check for errors, run echo \$? . If a 1 is returned, there are errors. If a 0 is returned, there were no errors.

Appendix: Oracle Trade Management System Expectations

This appendix describes the items expected by the Oracle Trade Management System.

Installation Scripts (elc_comp_post_htsupld.sql)

This script is for the RTM product only. This needs to be applied only after all static install scripts have been run, oga, tariff_treatment, quota_category, country_tariff_treatment and hts_headings scripts have all been run followed by running the htsupld.pc program. The last step is running this script. This script inserts the Expense and Assessment Cost Components. This script needs to be run once for each country of import that the client is using.

Note: This script is expecting two parameters to be passed in (the user will be prompted for the parameters). The first parameter is country ID, this is the Import Country. The second parameter is Currency Code, this is the code of the currency that corresponds to the entered Import Country. Most likely this script will be run using the Base Country and the Primary Currency as defined in the System Variables form.

The inserted components include:

- MPFXX (Merchandise Processing Fee XX) – This component is used to store Merchandise Processing Fee. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one component created, MPFUS, with a description of Merchandise Processing Fee US. This leaves the client with the ability to create additional MPF components for each of the countries that they intend to import into. This component is inserted with a Component Rate of 100 percent. This rate should be modified to be the appropriate rate for the Import Country. This component is also set up as an Always Default which means that it is defaulted to every Item/HTS combination.
- HMFXX (Harbor Maintenance Fee XX) – This component is used to store Harbor Maintenance Fee. In place of the XX will be the country code that is passed into the script. So if the Country is US, then there is one component created, HMFUS, with a description of Harbor Maintenance Fee US. This leaves the client with the ability to create additional HMF components for each of the countries that they intend to import into. This component is inserted with a Component Rate of 100 percent. This rate should be modified to be the appropriate rate for the Import Country.
- TDTYXX (Total Duty XX) – This component is used to store the total of the duty for each Item/HTS or Order/Item/HTS combination. It totals all duties, taxes, and fees within the Ordering dialog. This total is added together with the Total Expense and the Item's Cost to come up with the Total Estimated Landed Cost of the Item or Order/Item combination. This component should not be modified.

- VFDXX (Value For Duty XX) – This Computation Value Base (CVB) is used to store the value that duty should be calculated from. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one CVB created, VFDUS, with a description of Value for Duty US. This leaves the client with the ability to create additional VFD CVBs for each of the countries that they intend to import into. Upon insert here, this CVB will only have one detail, which is ORDCST (Order Cost). If the client needs additional expenses (we are making the assumption that only Expense components will make up Value for Duty) to be used in the Value For Duty, they need to be added to VFDXX through SQL Plus. All automatically inserted Assessment components with a Calculation Basis of Value will have VFDXX as the CVB.
- VFDXXXX (XX% of Value For Duty XX) – This component is used to store a percent of the CVB, Value For Duty. This is used in the case when you have an Item that is classified with multiple HTS codes. For example, a button-down shirt may have one HTS code for the cotton material that is 75 percent of the cost, and a second HTS code for the buttons that make up the other 25 percent. The duty components associated with the first HTS code would be need to be calculated from 75 percent of the entire Value for Duty. To accomplish this, the associated components would use VFD75XX as their CVB instead of VFDXX. The detail component would be 'VFD75XX' and would have a Component Rate of 75 and a CVB of VFDXX, therefore, the component VFD75XX would be 75% of the Value for Duty. More generically speaking, VFDXXXX will be the only detail in an inserted CVB called VFDXXXX, where the first XX is replaced with the percentage. In place of the second XX will be the country code that is passed into the script. So if the Country is US, then there will be one component created, VFD25US, with a description of 25% of Value for Duty US. This leaves the client with the ability to create additional VFD components for each of the countries that they intend to import into. The script will insert VFD25XX, VFD50XX, and VFD75XX, these are meant to be used as a guide if the client needs additional components with different percentages. These components should not be modified.
- DTYXXXX (DTYXXXX) – These components are used to calculate duty for each HTS code. In place of the first XX is the HTS code's Duty Component Code concatenated with an A, B, or C as needed for duty calculation. In place of the second XX is the country code that is passed into the script. So if the Country is US, then there is one component created, DTYXXUS, with a description of DTYXXUS. This leaves the client with the ability to create additional components for each of the countries that they intend to import into. The Import Country for these components is the country code of the Base Country that is defined on the System Options table. This component is inserted with a Component Rate of 100 percent. This rate is overwritten with the appropriate Tariff Treatment rate upon calculation within the Item and Ordering dialogs. These components should not be modified.
- DUTYXX(DUTYXX) – This component is used as a sub-total. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one component created, DUTYUS, with a description of DUTYUS. This leaves the client with the ability to create additional components for each of the countries that they intend to import into. It contains the sum of all DTYXXXX components each HTS code. This component has a CVB called DUTYXX that contains every 'DTYXXXX' component as its details. This component should not be modified.

- XXXXXXXX (XXXXXXX) – Fees and Taxes are created using a concatenation of information. The Component ID consists of the Fee or Tax Class Code concatenated with the Fee or Tax Component Code, and an A or B as needed for calculation, and then the import country. For example, there is an existing Fee Class Code (also referred to as Fee Type) which is 053, its Fee Component Code is 1, and importing into the US, so there is a component created that has an ID of 0531AUS. The descriptions are the same as the Component ID and can/should be modified to be clearer. Other than the description, these components should not be modified.
- ADXX (Anti-Dumping XX) – This component contains the Anti-Dumping charge for each Item/HTS code. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one component created, ADUS, with a description of Anti-Dumping US. This leaves the client with the ability to create additional components for each of the countries that they intend to import into. This component should not be modified.
- CVDXX (Countervailing Duty XX) – This component contains the Countervailing Duty charge for each Item/HTS code. In place of the XX will be the country code that is passed into the script. So if the Country is US, then there is one component created, CVDUS, with a description of Countervailing Duty US. This component should not be modified.

HTS Upload / Mass Update

There are several installation scripts that must be run prior to HTS Upload to populate the following tables. These are one-time installations upon implementation of the product and must be maintained by the client.

- ELC_COMP
- QUOTA_CATEGORY (through the quota_category.sql script)
- OGA (through the oga.sql script)
- COUNTRY_TARIFF_TREATMENT (via the country_tariff_treatment.sql script)
- HTS_CHAPTER (via the hts_headings.sql script)
- TARIFF_TREATMENT (through the tariff_treatment.sql script)

After the initial load of the HTS data from executing the HTS Upload program. One additional install script must be run to populate the following tables with additional information:

- ELC_COMP, CVB_HEAD, CVB_DETAIL (through the elc_comp_post_htsupld.sql script)

The initial load of HTS information using a Customs provided tape and subsequent execution of the HTS Upload program will populate and update the following tables:

- HTS
- HTS_TARIFF_TREATMENT
- HTS_OGA
- HTS_FEE
- HTS_TAX
- HTS_TT_EXCLUSIONS

The following tables need to be populated by the client, but will be updated through the HTS Upload program.

- HTS_AD
- HTS_CVD
- HTS_REFERENCE

The following tables need to be populated and maintained by the client:

- HTS_CHAPTER_RESTRAINTS

Calculation of Merchandise Processing Fee

This particular cost component is the only Cost Component that is calculated with a Min/Max Range for each Customs Entry. This range is defined on the MPF_MIN_MAX table (note: this table does not have a corresponding form and needs to be populated by the client via SQL Plus. In order to process MPF the MPF_MIN_MAX table must be populated for the import country or else the calculation function errors out during processing.). If a client does not use Merchandise Processing Fee, but has a similar component, they can use the MPF_MIN_MAX table and the MPFXX component to accomplish the same result. They simply need to change the Component Description and Rate. Within the Customs Entry dialog, MPFXX is defaulted in along with all other assessments that are associated with each Order/Item combination. Once associated with the Entry, MPF is recalculated and checked to see if the value falls within the Min/Max Range. If not, the value is modified to be within the range and then allocated across all of the items on the Entry. Because this value is being calculated by the system, the user is not allowed to modify the rate or value of any MPF components within the Customs Entry dialog.

Unit of Measure Conversions

The internal process that calculates and distributes MPF charges on-line requires Unit of Measure (UOM) conversions in multiple instances. If a particular UOM conversion is missing the processing stops and a message will be displayed indicating that there is insufficient UOM information to continue. If this should occur, you must exit the dialog that generated the error add the missing conversion information and re-enter the dialog for the MPF charges to be processed.

Customs Entry Ref. Status

There are 4 possible CE Ref. Statuses for each Customs Entry. They are Worksheet, Send, Downloaded, and Confirmed. In general when an Entry is created it is in Worksheet status. Once all of the necessary information has been added, the user is set the Status to Send, indicating that the Entry is ready to be sent to the Broker. That night in the nightly batch run, the Entry is downloaded to the Broker (cednld.pc). Once the download process is complete, the Status is automatically set to Downloaded; a user can never set the Status to this value manually. At that point once the user receives confirmation from the Broker, makes any necessary changes, and is sure that the information is correct, they can set the CE Ref. Status to 'Confirmed'. From that point on the Status cannot be changed, however most of the fields on the CE Header form remain editable. All information on the CE Shipment form is view only. Also, all information on the CE Order/Item form is view only except for the Cleared Quantity, Cleared Quantity UOM, Apply button, and Comments fields. And finally all information in the CE Charges form will be view only as well.

Since some clients may prefer not to download their Entries to a Broker, the user will have the ability to set the CE Ref. Status from Worksheet directly to Confirmed.

Customs Entry Totals

The following describes customs entry totals.

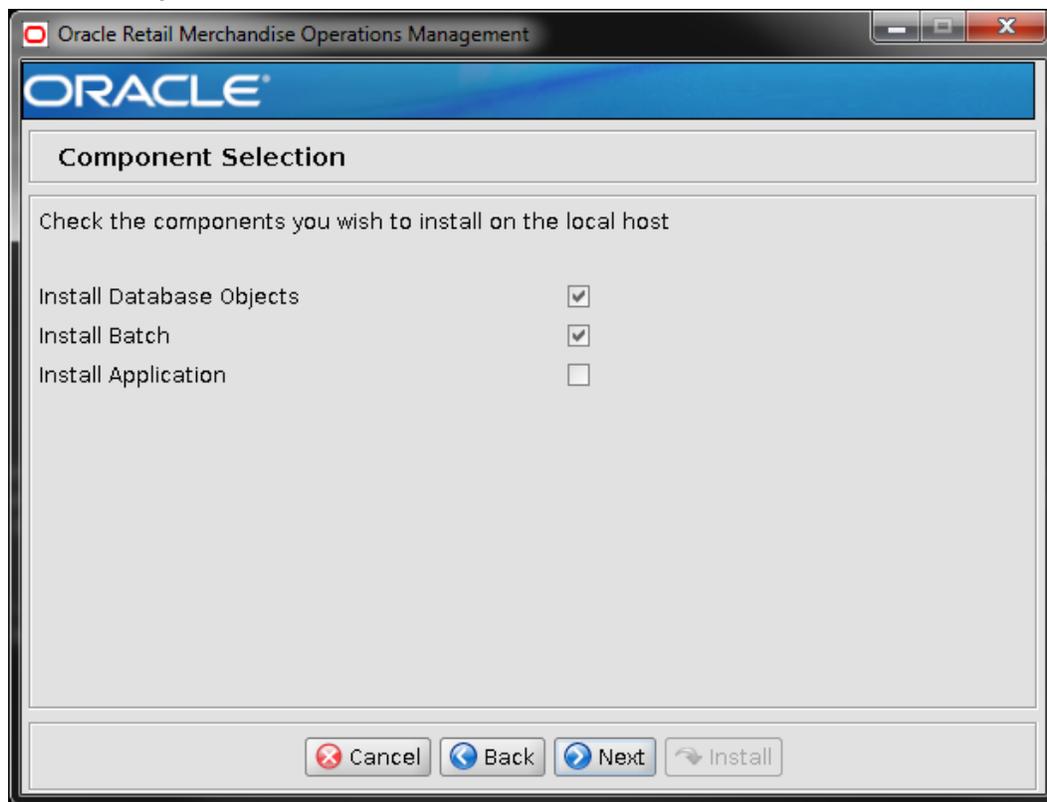
- Total Duty contains the sum of the duty charges (any component beginning with DTY) for each item times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Taxes contains the sum of the tax charges (any component beginning with a tax type (see attached document for a description of taxes)) for each item times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Other contains the sum of all other charges (including fees) for each item times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total VFD contains the Value for Duty (which can be made up of order cost plus other dutiable expenses such as selling commission, royalties, etc.) times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Est. Assessments contains the sum of the estimated duty/fees/taxes for each item, calculated from the Purchase Order/Item HTS Assessments, times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Act. Assessments contain the sum of the Total Duty, Total Taxes, and Total Other values.

Appendix: RMS Database Schema and Batch Installation Screens

You need the following details about your environment for the installer to successfully create the RMS database schema and install the RMS batch programs. Depending on the options you select, you may not see some screens or fields.

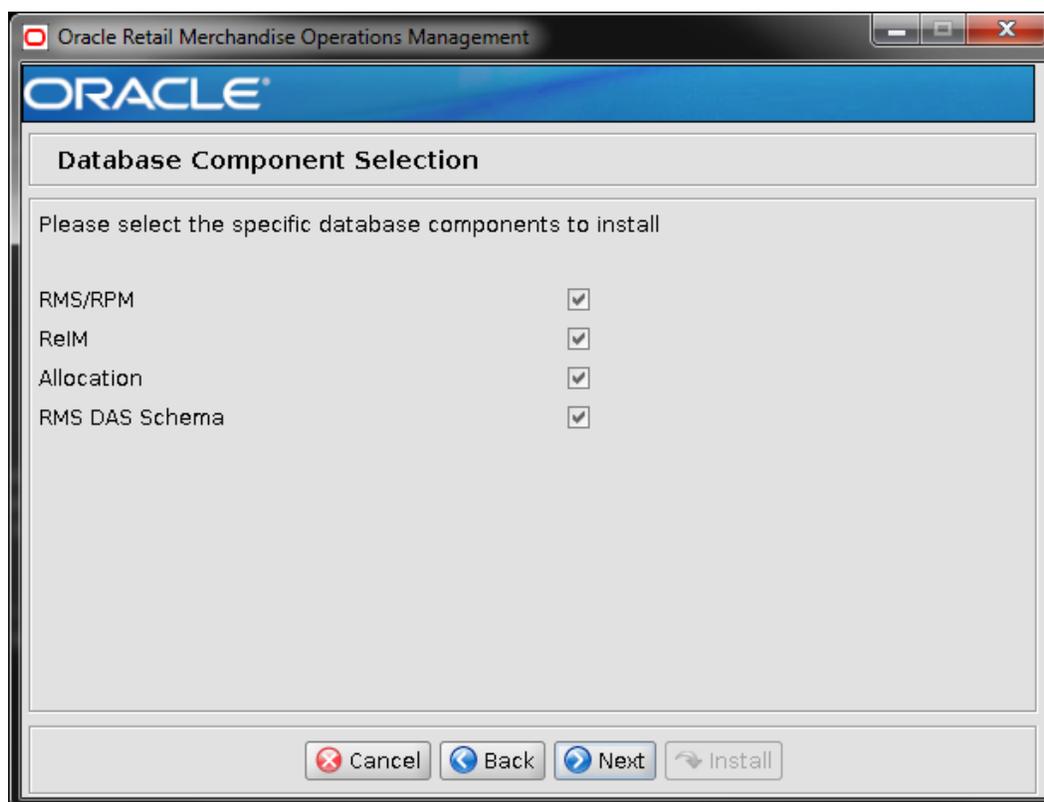
The RMS database schema installation also includes the option to install the database schema objects for the ReIM and Allocation products. The RPM database schema objects will be included with RMS.

Screen: Component Selection

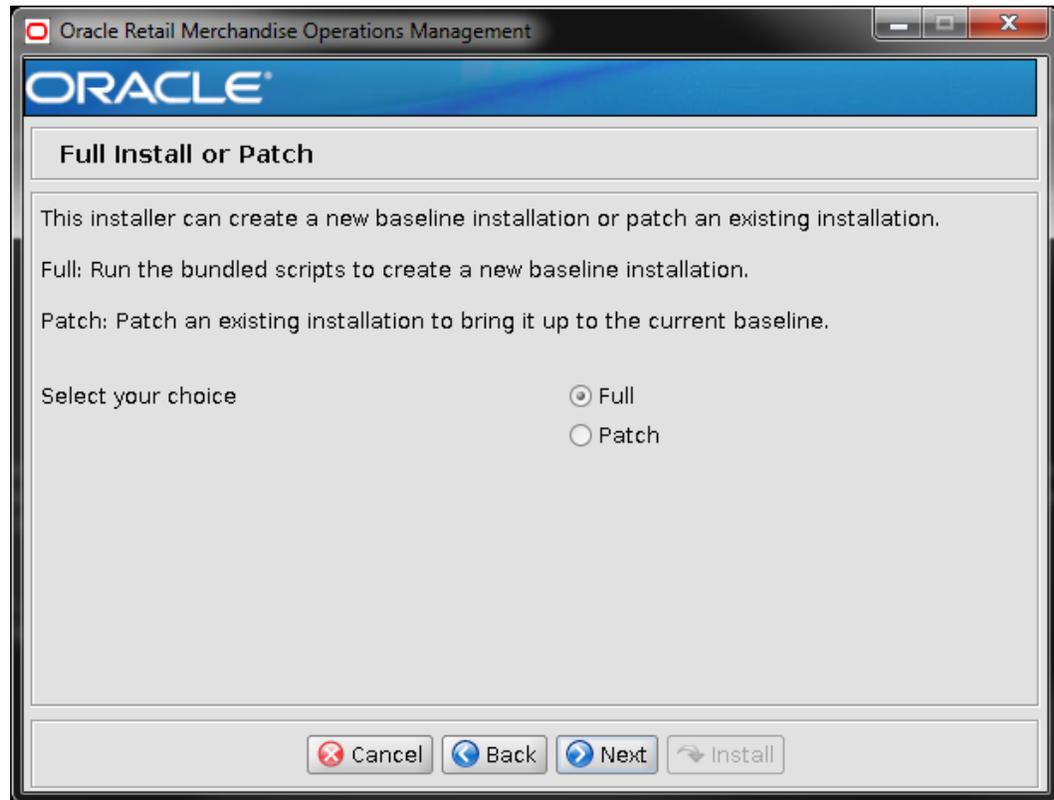


Field Title	Component Selection
Field Description	Select the RMS component(s) you would like to install. Multiple components may be selected. You will not be able to install a component if the preinstall check for that component has failed. Subsequent screens may or may not be displayed based on this choice.

Screen: Database Component Selection



Field Title	Database Component Selection
Field Description	By default, the RMS database schema installer creates database objects for RMS/ReSA/RTM and RPM. Optionally, the database objects for ReIM, Allocation and/or RMS DAS Schema may be installed at the same time or later. Subsequent screens may or may not be displayed based on this choice.
Note	RMS DAS Schema is applicable if setting up DAS.

Screen: Full Install or Patch

Field Title	Full Install or Patch
Field Description	The installer can create either the full baseline schema or upgrade an existing installation. To install a new instance of RMS 14.1 release, select Full. If upgrading from 14.0 or 14.0.1, please select Patch. Subsequent screens may or may not be displayed based on this choice.
Example	Full

Screen: Host Details

Oracle Retail Merchandise Operations Management

ORACLE

Host Details

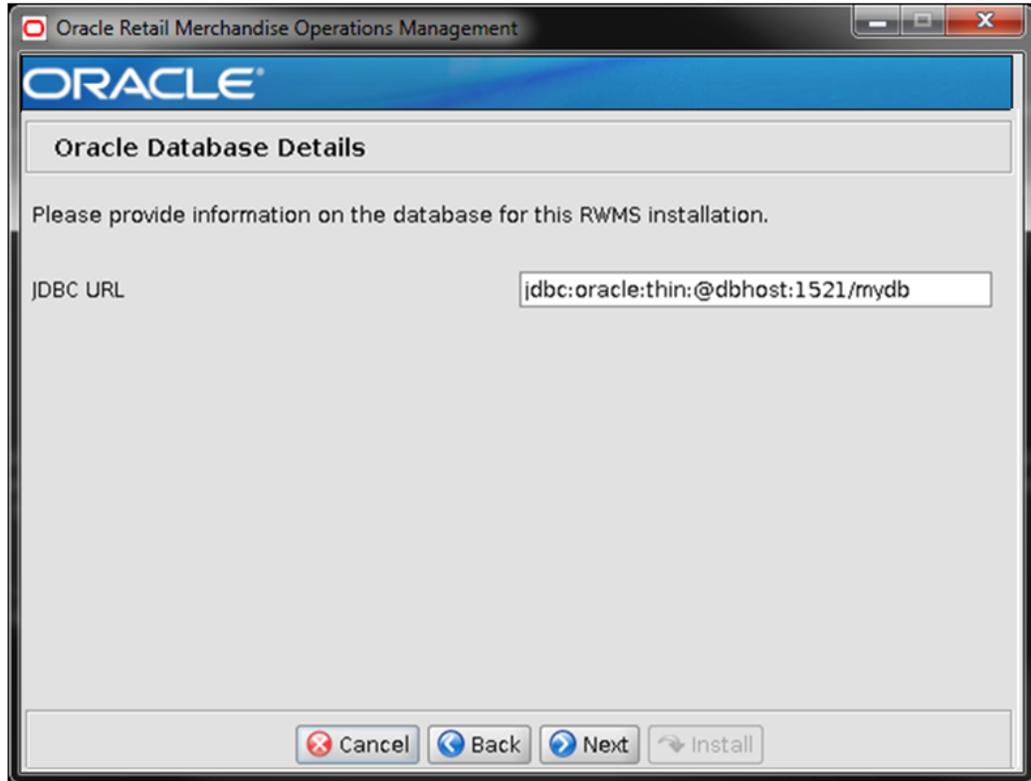
Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname

Cancel Back Next Install

Field Title	Hostname
Field Description	Provide the hostname of the Oracle Database Server
Example	dbhostname

Screen: Oracle Database Details



Field Title	JDBC URL
Field Description	URL used by the installer to access the RMS database schema. See Appendix: URL Reference for expected syntax.
Examples	<p>Standard Thin Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydb</code></p> <p>Standard Pluggable DB Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydbservice</code></p> <p>Standard OCI Connection: <code>jdbc:oracle:oci:@mydb</code></p> <p>RAC connection: <code>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname2)(PORT=1521))(LOAD_BALANCE=yes))(CONNECT_DATA=(SERVICE_NAME=mydb)))</code></p>

Screen: RMS Database Schema Details

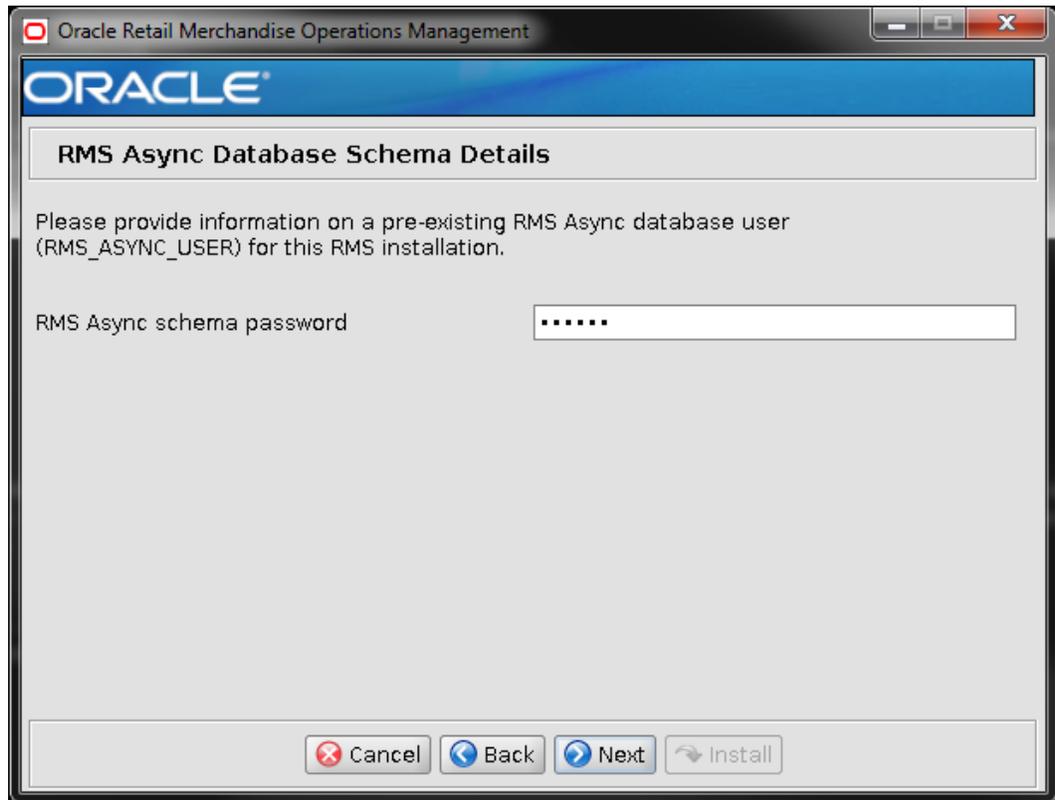
Field Title	RMS schema
Field Description	Provide the RMS database user here. The installer logs into the database as this user to create the RMS schema and uses it to compile RMS batch. This user must already exist in the database when the RMS database schema installer is run.
Example	rms01

Field Title	RMS schema password
Field Description	Database password for the RMS Schema Owner.

Field Title	RMS Oracle SID
Field Description	Oracle system identifier for the database where RMS will be installed
Example	mydb

The database settings provided are validated by the installer when you advance to the next screen.

Screen: RMS Async Database Schema Details



Field Title	RMS Async schema password
Field Description	Database password for the RMS Async schema – RMS_ASYNC_USER. This user must already exist in the database when the RMS database schema installer is run.

The database settings provided are validated by the installer when you advance to the next screen.

Screen: Allocation Database Schema Details

Field Title	Alloc schema
Field Description	Provide the Allocation database user here. The installer logs into the database as this user to create the Allocation schema objects. This user must already exist in the database when the database schema installer is run.
Example	alloc01

Field Title	Alloc schema password
Field Description	Database password for the Allocation database user.

The database settings provided are validated by the installer when you advance to the next screen.

Screen: RMS DAS Database Schema Details

Field Title	RMS DAS schema
Field Description	Provide the RMS DAS database user here. The installer logs into the database as this user to create the DAS schema objects. This user must already exist in the database when the database schema installer is run.
Example	rms01das

Field Title	RMS DAS schema password
Field Description	Database password for the RMS DAS schema.

Field Title	RMS DAS schema Oracle SID
Field Description	Oracle system identifier for the database where RMS DAS schema will be installed.
Example	mydasdb
Note	The DAS Schema must be created in a different database instance than that of RMS schema.

The database settings provided are validated by the installer when you advance to the next screen.

Note: The next 18 screens are only shown for a FULL installation, and not for UPGRADE installation.

Screen: RMS Primary Country



Field Title	Primary Country
Field Description	Choose your primary country from the dropdown list provided.
Example	UNITED STATES (US)

Screen: RMS Primary Currency

Oracle Retail Merchandise Operations Management

ORACLE

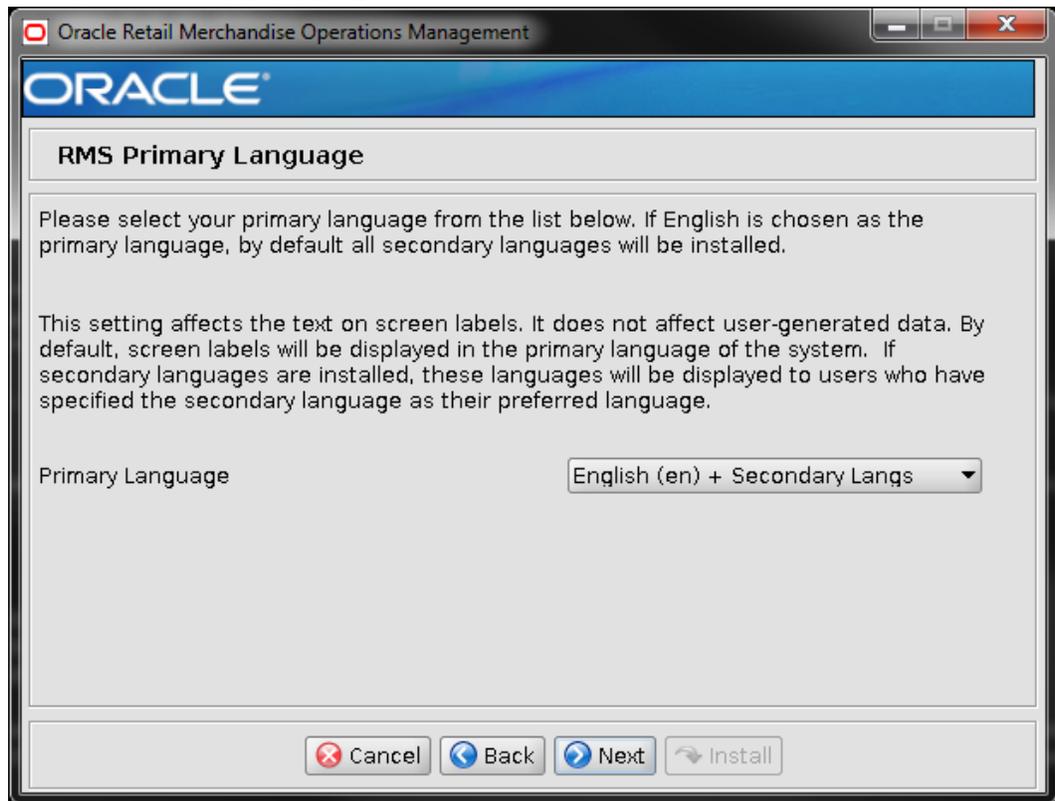
RMS Primary Currency

This will be the base currency for the merchandising system. The primary currency is used throughout the system in various ways. For one, any conversion between currencies will utilize the primary currency. For example, if currency A is the primary currency and the system is converting from currency B to currency C it will first convert currency B to currency A, then currency A to currency C. As a result, all currency exchange rates reflect the rate between the non-primary currency and the primary currency.

Primary Currency US Dollar (USD) ▼

Field Title	Primary Currency
Field Description	Choose your primary currency from the dropdown list provided.
Example	US Dollar (USD)

Screen: RMS Primary Language



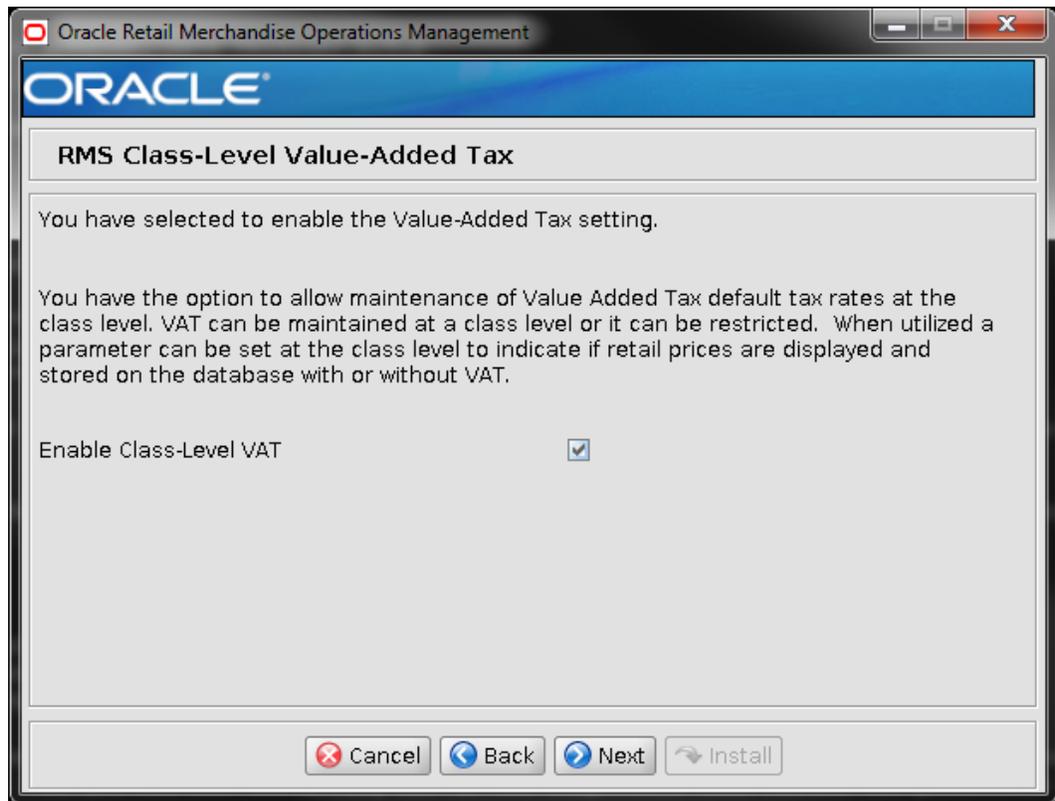
Field Title	Primary Language
Field Description	Choose your primary language from the dropdown list provided. If English is chosen, by default, all secondary languages will be installed. If you choose a non-English primary language, no secondary languages will be installed.
Example	English (en) + Secondary Languages

Screen: RMS Default Tax Type

Field Title	Default Tax type
Field Description	Select the tax type that will be used with the system. SVAT: Simple Value-Added Tax (VAT information is configured in RMS) SALES: Sales and Use Tax If VAT is enabled, then select SVAT. For a configuration with only Sales Tax, Select SALES.

Note: The RMS Class level VAT screen is only shown if SVAT is selected on the Default Tax Type.

Screen: RMS Class-Level Value-Added Tax



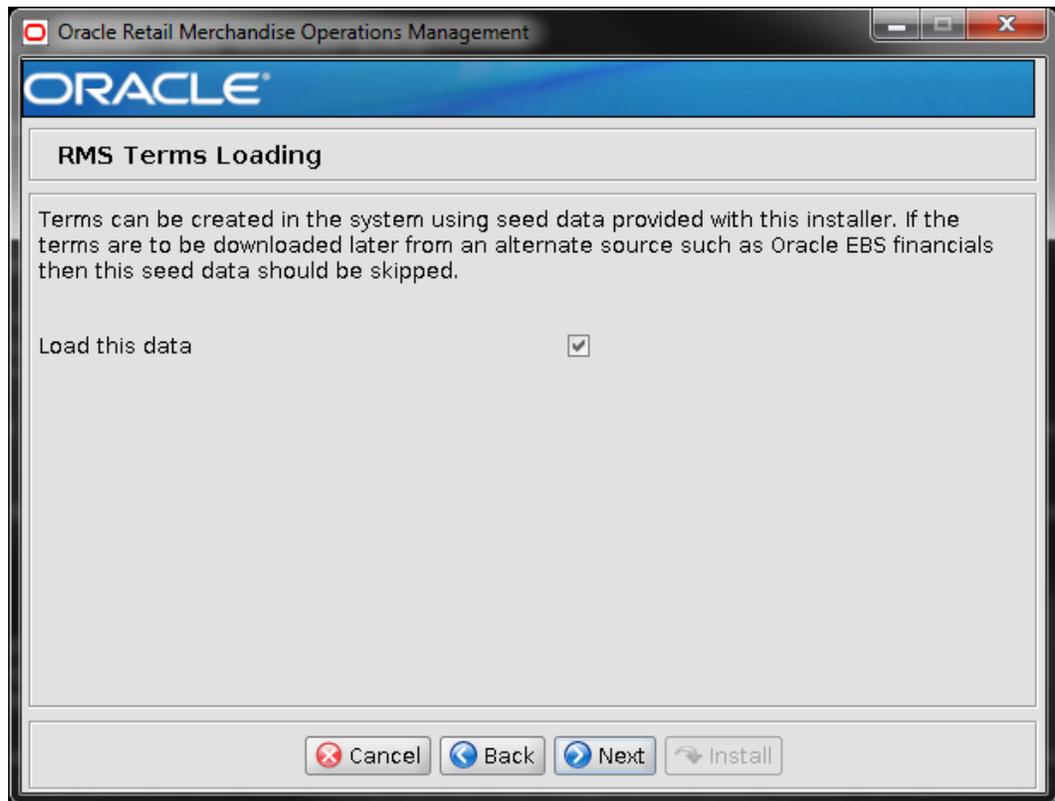
Field Title	Enable Class-Level VAT
Field Description	Check the box to allow tax rates to be maintained at the class level. Leave the box unchecked to restrict tax rates.

Screen: RMS Supplier Sites

The screenshot shows a window titled "Oracle Retail Merchandise Operations Management" with the Oracle logo at the top. Below the logo is a section titled "RMS Supplier Sites". The text in this section reads: "This parameter is used to indicate if a supplier hierarchy is being utilized in the system. If this is enabled a supplier can be further differentiated by different physical 'sites'. If this is disabled then a flat supplier structure will be used so every supplier corresponds to a single site." Below this text is a checkbox labeled "Enable Supplier Sites" which is checked. At the bottom of the window are four buttons: "Cancel", "Back", "Next", and "Install".

Field Title	Enable Supplier Sites
Field Description	Indicates if a supplier hierarchy is being utilized in the system.

Screen: RMS Terms Loading



Field Title	Load this data
Field Description	Terms data is provided with the RMS release. Select this option to insert it into the schema. If data will be pulled from another system such as EBS financials then do not select this option.

Screen: RMS Calendar Type

Oracle Retail Merchandise Operations Management

ORACLE

RMS Calendar Type

A "4-5-4" calendar is one containing reporting periods of 4-weeks, 5-weeks, and 4-weeks. Four of these reporting periods represent a reporting year. A "Standard" calendar indicates that a typical 12-month calendar is being used for financial reporting.

Select Calendar Type

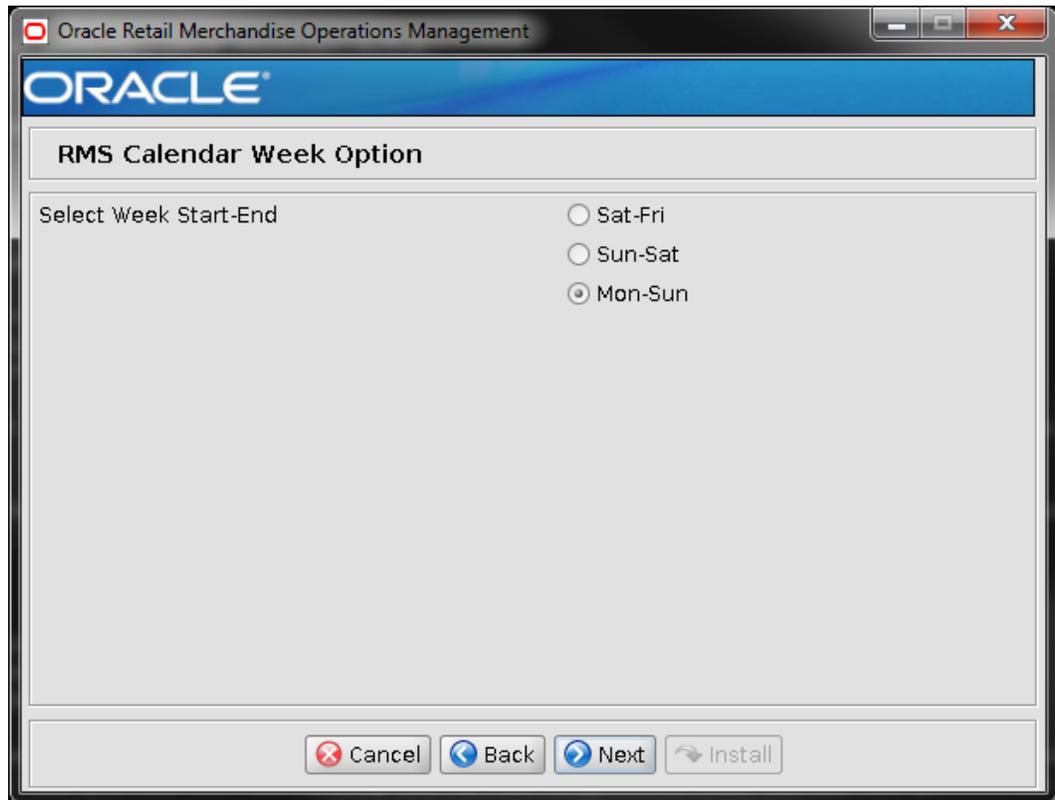
454 Calendar

Standard Calendar

Cancel Back Next Install

Field Title	Select Calendar Type
Field Description	Choose the type of calendar to use.

Screen: RMS Calendar Week Option



Field Title	Select Week Start-End
Field Description	Select the range that defines the first and last days of the week.

Screen: RMS Calendar VDate

Oracle Retail Merchandise Operations Management

ORACLE

RMS Calendar VDate

This should contain the first date the RMS system will be in operation. The vdate represents the business date within RMS and it is also leveraged outside of RMS by some other satellite applications. VDate must be at least one month after the RMS calendar start date.

Date format is dd-MMM-yyyy (Example: 01-MAR-2008)

VDate

Cancel Back Next Install

Field Title	VDate
Field Description	Enter the first date the RMS System will be in operation. The format dd-MMM-yyyy must be used.
Example	23-Oct-2015

Screen: HTS Tracking Level



Field Title	HTS Tracking Level
Field Description	Select the basis for HTS tariffs and fees. The options are either the item's country of manufacturer or the item's country of sourcing.

Screen: Data Level Security

Oracle Retail Merchandise Operations Management

ORACLE

Data Level Security

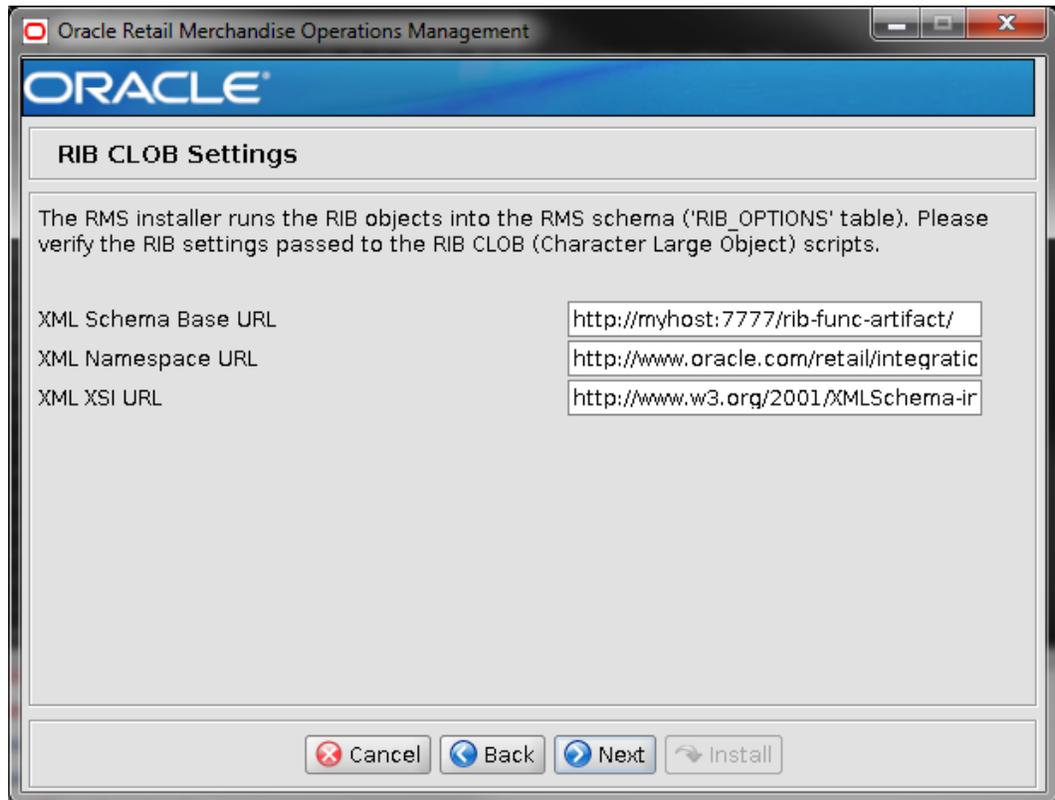
Data level security provides the option to restrict user's access to specific data within the merchandising system based on merchandise hierarchy or organizational hierarchy.

Enable Data Level Security?

Cancel Back Next Install

Field Title	Enable Data Level Security?
Field Description	Indicates if data level security is being utilized in the system.

Screen: RIB CLOB Settings

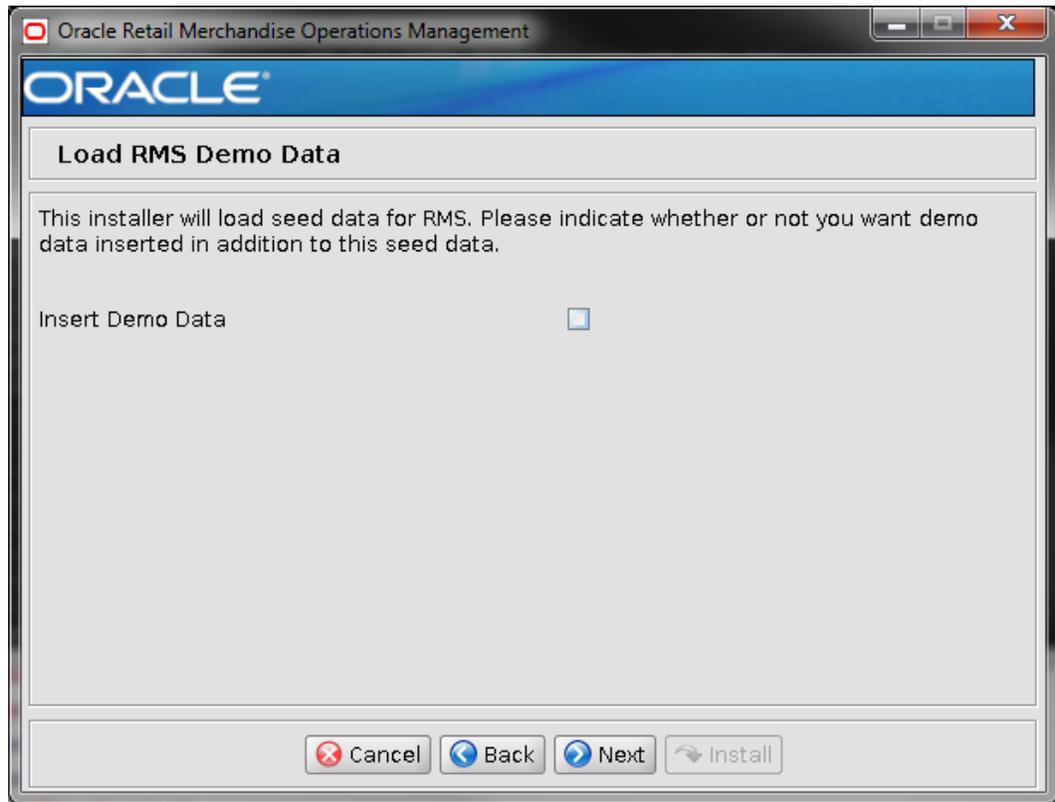


Field Title	XML Schema Base URL
Field Description	URL for the RIB object schema definition.
Example	http://myhost:7777/rib-func-artifact/ Note: Here, the host and ports must be from RIB that is probably installed later.

Field Title	XML Namespace URL
Field Description	URL for the RIB object namespace.
Example	http://www.oracle.com/retail/integration/base/bo

Field Title	XML XSI URL
Field Description	URL for the XML schema instance.
Example	http://www.w3.org/2001/XMLSchema-instance

Screen: Load RMS Demo Data



Field Title	Insert Demo Data
Field Description	The Installer by default loads seed data for RMS. Check the box to insert RMS demo data in addition to seed data.
Note	Demo data should not be installed in production environments. See the warning screen below for more details.

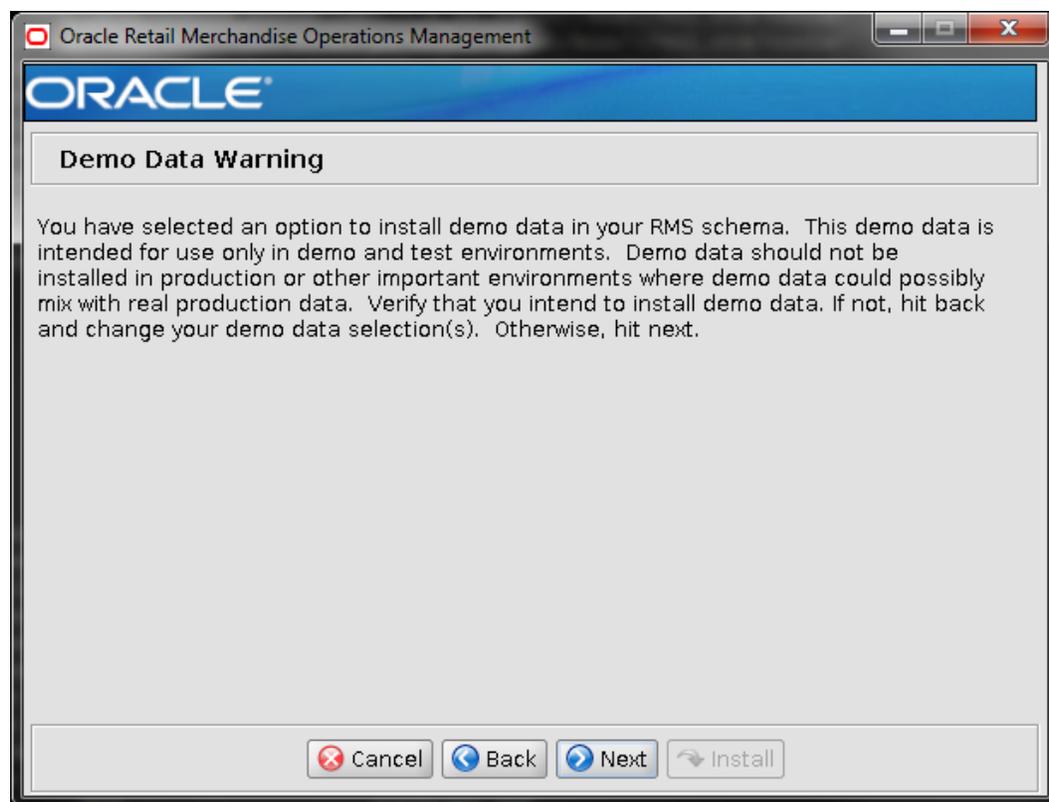
Screen: Load ReIM Demo Data

Note: The Load ReIM Demo Data screen is only shown if Insert Demo Data is checked in the previous screen and ReIM is selected in the beginning of the installation.

Field Title	Insert ReIM Demo Data
Field Description	Check the box to insert ReIM demo data.
Note	Demo data should not be installed in production environments. See the warning screen below for more details.

Screen: Demo Data Warning

Note: This screen is shown only if a Demo Data option is selected in the previous screens. Please read the Warning carefully.



Screen: RMS Demo Data Schema Details

Oracle Retail Merchandise Operations Management

ORACLE

RMS Demo Data Schema Details

Please provide information on a pre-existing database user for RMS Demo Data. The installer will authenticate as this user and create the demo data in this schema.

Demo Data schema: rms01demo

Demo Data schema password:

Cancel Back Next Install

Field Title	Demo Data schema
Field Description	Schema that will be used to insert demo data into the RMS database.
Example	rms01demo

Field Title	Demo Data schema password
Field Description	Password for the demo data schema.

Screen: RMS Demo Data – Number of Items

Oracle Retail Merchandise Operations Management

ORACLE

RMS Demo Data - Number of Items

Select number of demo items to create

Number of demo items

Cancel Back Next Install

Field Title	Number of demo items
Field Description	The number of demo data items to create.
Example	15

Screen: RMS Demo Data – Transaction Level

The screenshot shows a window titled "Oracle Retail Merchandise Operations Management" with the Oracle logo at the top. Below the logo is a header "RMS Demo Data - Transaction Level". The main content area contains the following text:

Please select a value to use for the transaction levels of the demo items being created. This is not a permanent setting in the system. Only the demo data created by this installer will be affected by this choice.

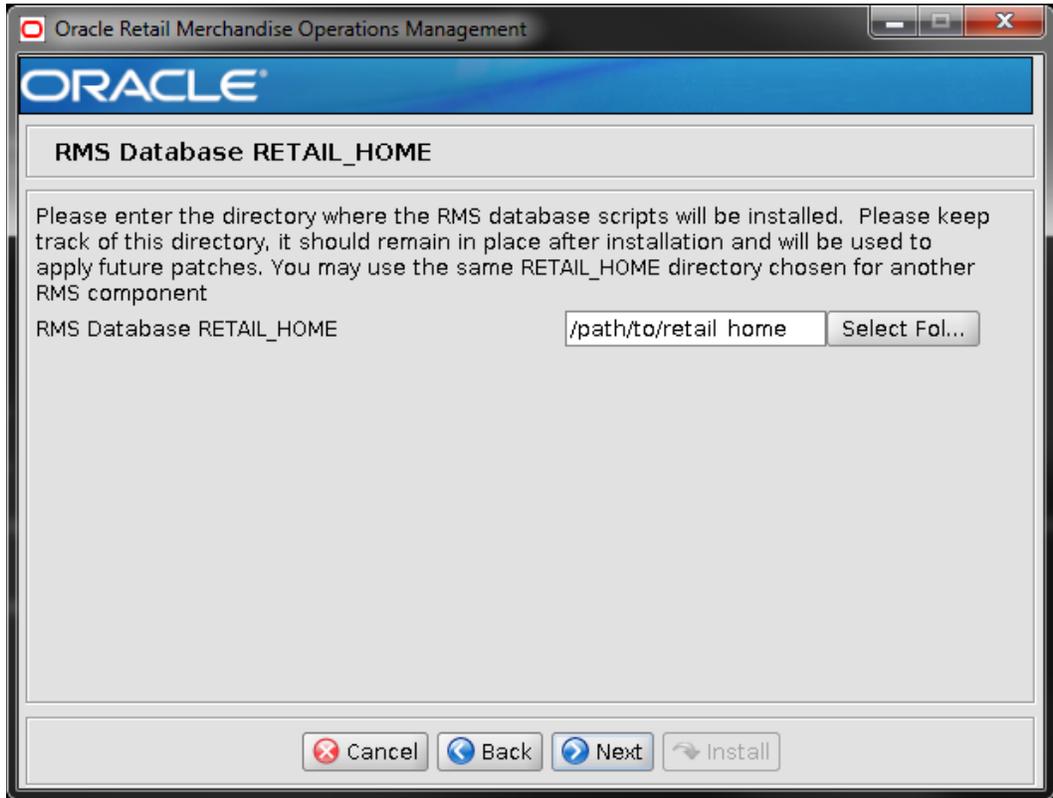
Transaction Level

- 1 (Line)
- 2 (Line Extension)
- 3 (Variant)

At the bottom of the window, there are four buttons: "Cancel", "Back", "Next", and "Install".

Field Title	Transaction Level
Field Description	Value to use for the transaction level of the demo items being created.
Example	1 (Line)

Screen: RMS Database RETAIL_HOME



Field Title	RMS DB RETAIL_HOME
Field Description	The location where the RMS Database Files are stored by the installer. This location will be used during the subsequent patching of RMS, and will contain the ORPatch utility.
Example	/path/to/retail_home
Note	If you have selected an existing RETAIL_HOME, and it has been configured to run other components than the ones you have selected for this installation, those components will also be installed regardless of what you selected on the Component Selection screen.

Screen: RMS Batch RETAIL_HOME

Oracle Retail Merchandise Operations Management

ORACLE

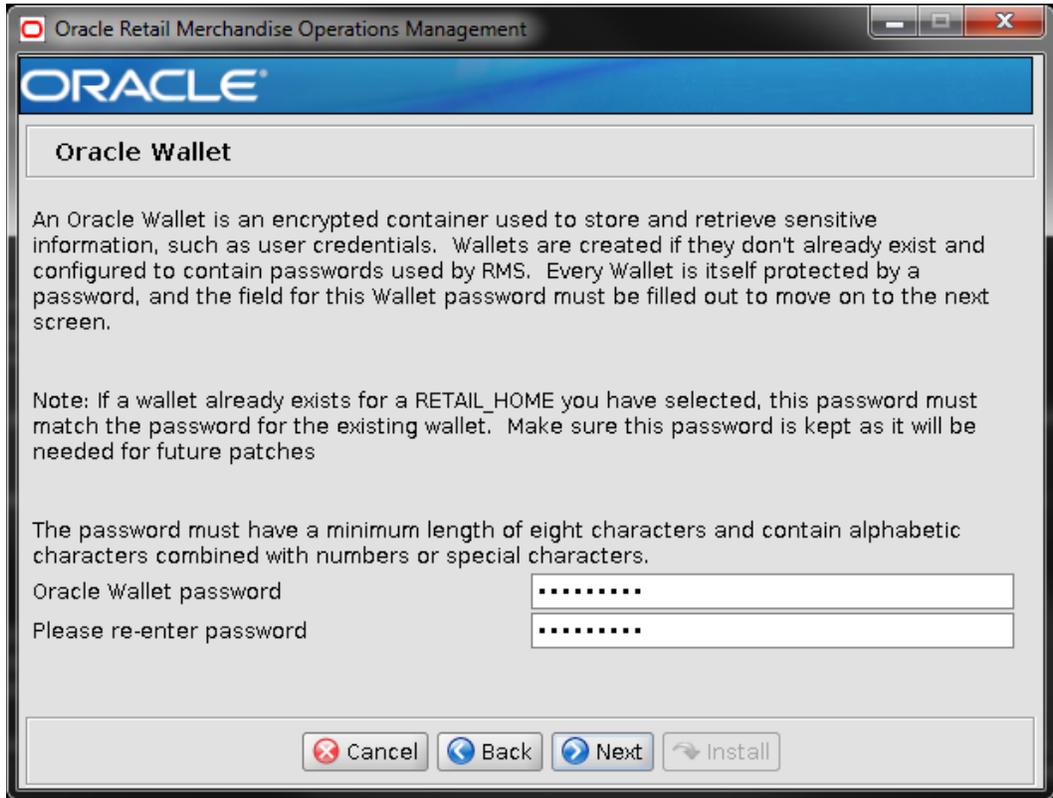
RMS Batch RETAIL_HOME

Please enter the directory where RMS Batch will be installed. Please keep track of this directory, it should remain in place after installation and will be used to apply future patches. You may use the same RETAIL_HOME directory chosen for another RMS component

RMS Batch RETAIL_HOME

Field Title	RMS Batch RETAIL_HOME
Field Description	This is the Batch Installation Directory, the location where RMS Batch Files will be installed along with the ORPATCH utility. This can be the same RETAIL_HOME that was used for another component.
Example	/path/to/retail_home
Note	If you have selected an existing RETAIL_HOME, and it has been configured to run other components than the ones you have selected for this installation, those components will also be installed regardless of what you selected on the Component Selection screen.

Screen: Oracle Wallet



Field Title	Oracle Wallet
Oracle Wallet Password	This is the password for the wallet that will store the credentials used during the RMS installation. If you have selected an existing RETAIL_HOME in the previous screens, you will need to enter the password that was used for the wallet in that RETAIL_HOME.
Note	Make sure this password is kept as it will be needed for future upgrades.

Appendix: RMS Application Installer Screens

Screen: Component Selection

Oracle Retail Merchandise Operations Management

ORACLE

Component Selection

Check the components you wish to install on the local host

Install Database Objects

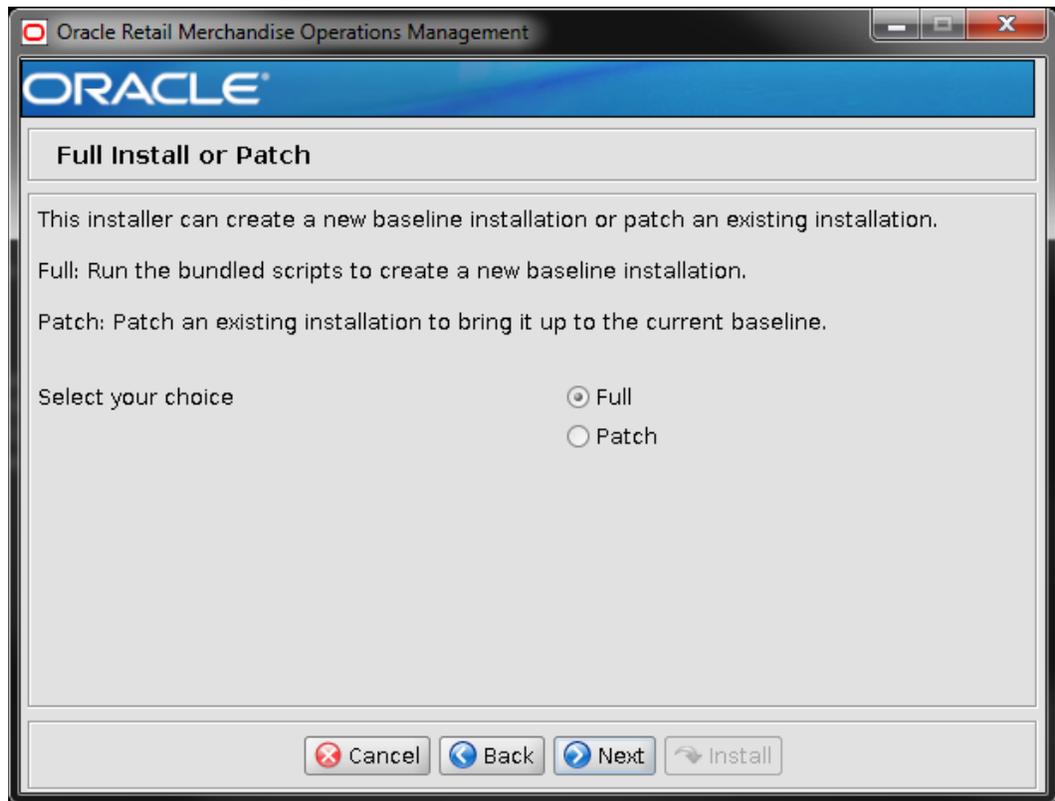
Install Batch

Install Application

Cancel Back Next Install

Field Title	Component Selection
Field Description	Select the RMS component(s) you would like to install. Multiple components may be selected. You will not be able to install a component if the preinstall check for that component has failed. Subsequent screens may or may not be displayed based on this choice.

Screen: Full Install or Upgrade



Field Title	Full Install or Patch
Field Description	The option selected on this page has no impact on the RMS forms installation. All Application installations will be full installations.
Example	Full

Screen: Host Details

Oracle Retail Merchandise Operations Management

ORACLE

Host Details

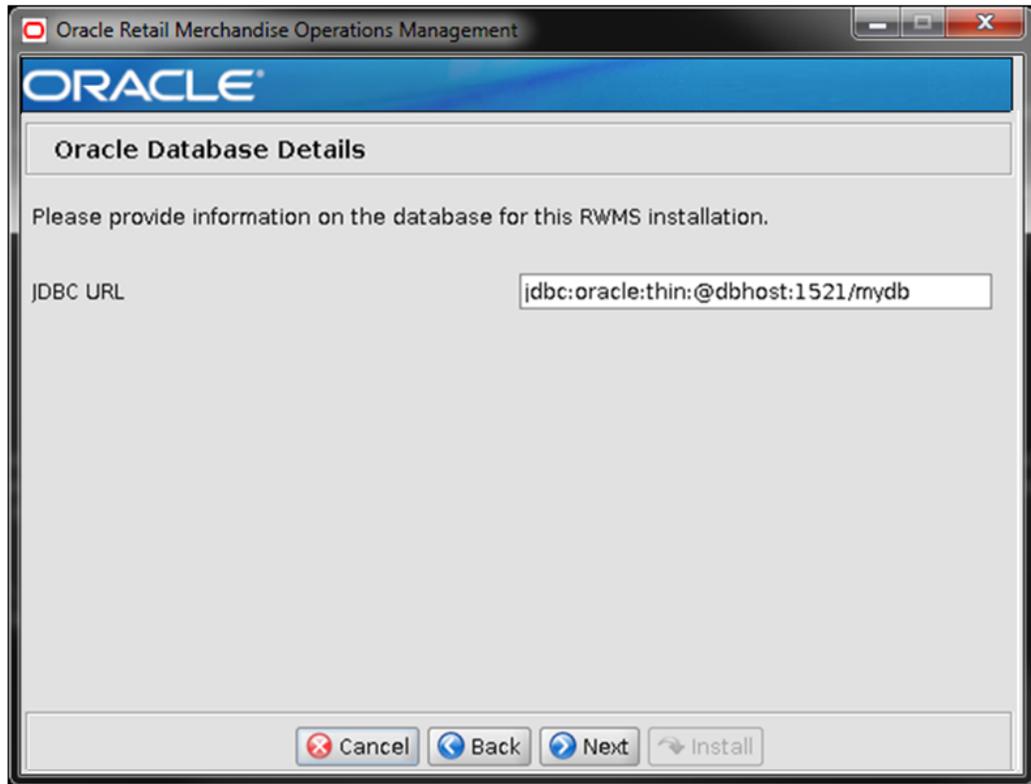
Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname

Cancel Back Next Install

Field Title	Hostname
Field Description	Provide the hostname where the RMS Application is being installed.
Example	Apphostname

Screen: Oracle Database Details



Field Title	JDBC URL
Field Description	URL used by the installer to access the RMS database schema. See Appendix: URL Reference for expected syntax.
Examples	<p>Standard Thin Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydb</code></p> <p>Standard Pluggable DB Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydbservice</code></p> <p>Standard OCI Connection: <code>jdbc:oracle:oci:@mydb</code></p> <p>RAC connection: <code>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname2)(PORT=1521))(LOAD_BALANCE=yes))(CONNECT_DATA=(SERVICE_NAME=mydb)))</code></p>

Screen: RMS Database Schema Details

Oracle Retail Merchandise Operations Management

ORACLE

RMS Database Schema Details

Please provide information on a pre-existing database user for this RMS installation.

RMS schema: rms01

RMS schema password:

RMS Oracle SID: mydb

Buttons: Cancel, Back, Next, Install

Field Title	RMS schema
Field Description	Provide the RMS database user here. This is the same username that was used during the RMS Database Schema Installation.
Example	rms01

Field Title	RMS schema password
Field Description	Database password for the RMS Schema Owner. This is the same password that was used during the RMS Database Schema Installation.

Field Title	RMS Oracle SID
Field Description	Oracle system identifier for the database the RMS application will be installed against. This is the same Oracle SID that was used during the RMS Database Schema Installation.
Example	mydb

The database settings provided are validated by the installer when you advance to the next screen.

Screen: Installation Name

Field Title	Application Installation Name
Field Description	The value used to configure and identify a specific forms installation. The Forms URLs for this installation will contain this value.
Example	rmsinst

Screen: WebLogic Administrative Details

Oracle Retail Merchandise Operations Management

ORACLE

Weblogic Administrative Details

Enter the administrative user and password for the Weblogic Server to which the application will be deployed.

Weblogic Admin port: 7001

Weblogic Admin User: weblogic

Weblogic Admin Password:

Please re-enter password:

Enable SSL for RMS? Yes No

Note: enabling SSL requires that security certificates have been configured and installed for this WebLogic domain. The AdminServer and all managed servers must then be configured to use SSL.

Cancel Back Next Install

Field Title	WebLogic Admin Port
Field Description	Listen port for the WebLogic Admin server.
Example	7001

Field Title	WebLogic Admin User
Field Description	Username of the admin user for the WebLogic instance to which the Oracle Forms application is already deployed.
Example	weblogic

Field Title	WebLogic Admin Password
Field Description	Password for the WebLogic admin user. You chose this password when you created the WebLogic instance.

Field Title	Enable SSL for RMS?
Field Description	Choose Yes to install RMS Forms using a WebLogic environment configured to use SSL. In this case, SSL must be configured and the ports must be enabled for the AdminServer and Oracle Forms managed servers. Choose No to install using a WebLogic environment configured without SSL. In this case the non-SSL ports must be enabled for the AdminServer and for the Oracle Forms managed servers.

Screen: Webhelp Installation Details

Oracle Retail Merchandise Operations Management

ORACLE

Webhelp Installation Details

RMS webhelp provides enhanced accessibility and usability of product documentation. The installation of webhelp requires a running Weblogic managed server instance. If a managed server has not been configured or is not running, please see the documentation on pre-installation setup requirements.

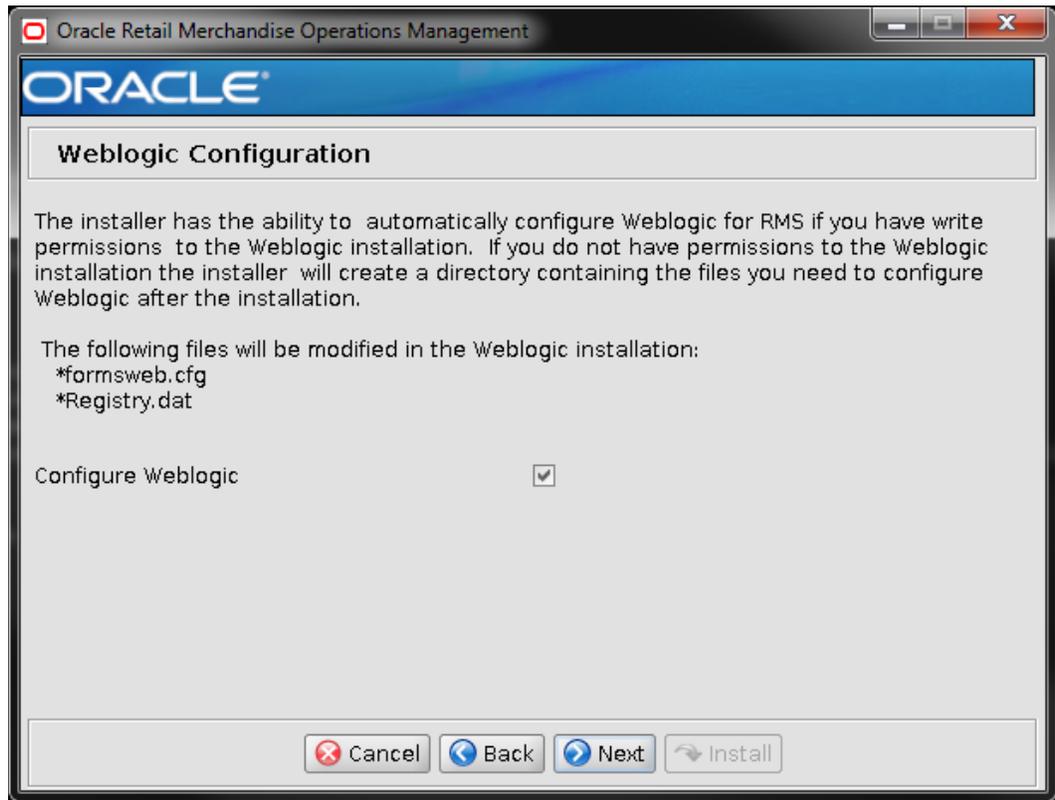
Enter the Weblogic managed server for RMS webhelp.

RMS Help Server

Cancel Back Next Install

Field Title	RMS Help Server
Field Description	The WebLogic managed server that was created for the RMS Webhelp application.
Example	RMS_HELP

Screen: WebLogic Configuration



Field Title	Configure WebLogic
Field Description	Make the necessary configurations to the WebLogic server to be able to run RMS forms. If you choose no, the configured WebLogic files will be generated by the installer, but should be applied to WebLogic manually.

Screen: RMS Application RETAIL_HOME

Oracle Retail Merchandise Operations Management

ORACLE

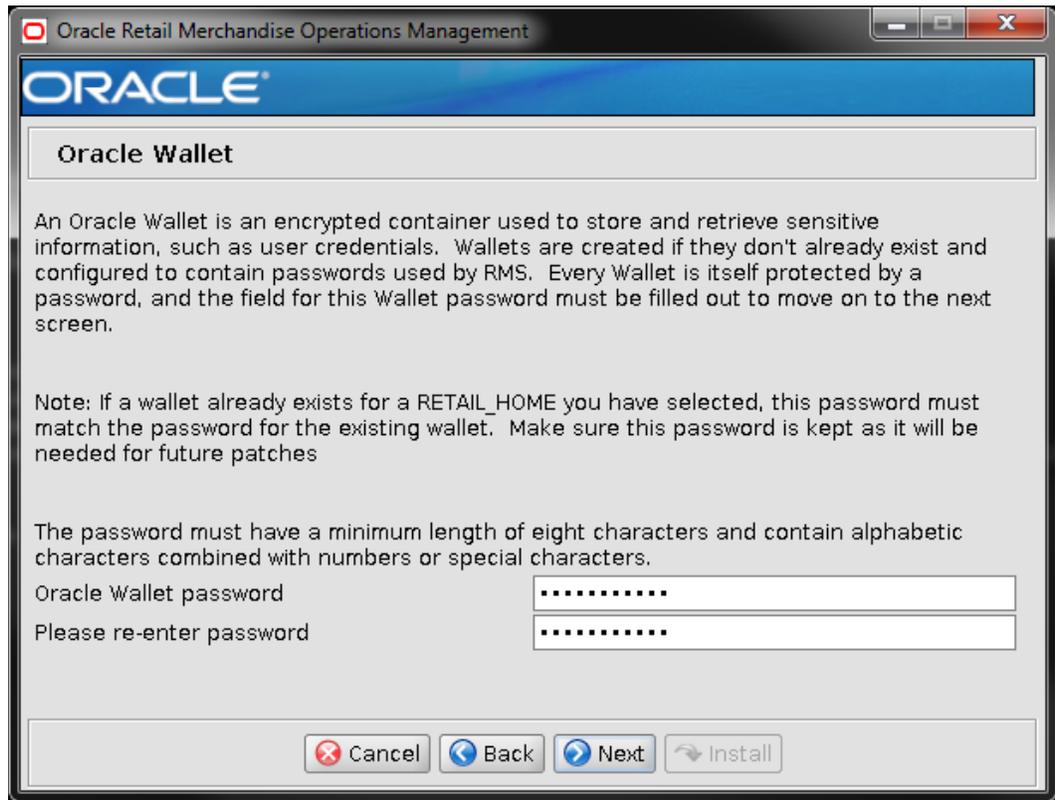
RMS Application RETAIL_HOME

Please enter the directory where RMS Application forms will be installed. Typically the RMS forms installation directory is located outside of the Weblogic installation. Please keep track of this directory, it should remain in place after installation and will be used to apply future patches. You may use the same RETAIL_HOME directory chosen for another RMS component

RMS Application RETAIL_HOME

Field Title	RMS Application RETAIL_HOME
Field Description	The location where the RMS Application (toolset, forms and reports) will be installed along with the ORPATCH utility. The RMS Forms will be installed in a subdirectory of this directory, named base.
Example	/path/to/retail_home
Note	If you have selected an existing RETAIL_HOME, and it has been configured to run other components than the ones you have selected for this installation, those components will also be installed regardless of what you selected on the Component Selection screen.

Screen: Oracle Wallet



Field Title	Oracle Wallet password
Field Description	This is the password for the wallet that will store the credentials used during the RMS installation. If you have selected an existing RETAIL_HOME in the previous screens, you will need to enter the password that was used for the wallet in that RETAIL_HOME.
Note	Make sure this password is kept as it will be needed for future upgrades.

Appendix: RMS Analyze Tool

It may be desirable to see a list of the files that will be updated by a patch, particularly if files in the environment have been customized. The installer has an ‘analyze’ mode that will evaluate all files in the patch against the environment and report on the files that will be updated based on the patch. See the section “Analyzing the Impact of a Patch” in the chapter “RMS Patching Procedures” for more details.

Run the RMS Analyze Tool

1. Log onto the server as a user with access to the RETAIL_HOME for the installation you want to analyze.
2. Change directories to STAGING_DIR/rms/installer. STAGING_DIR is the location where you extracted the installer.
3. Set and export the following environment variables.

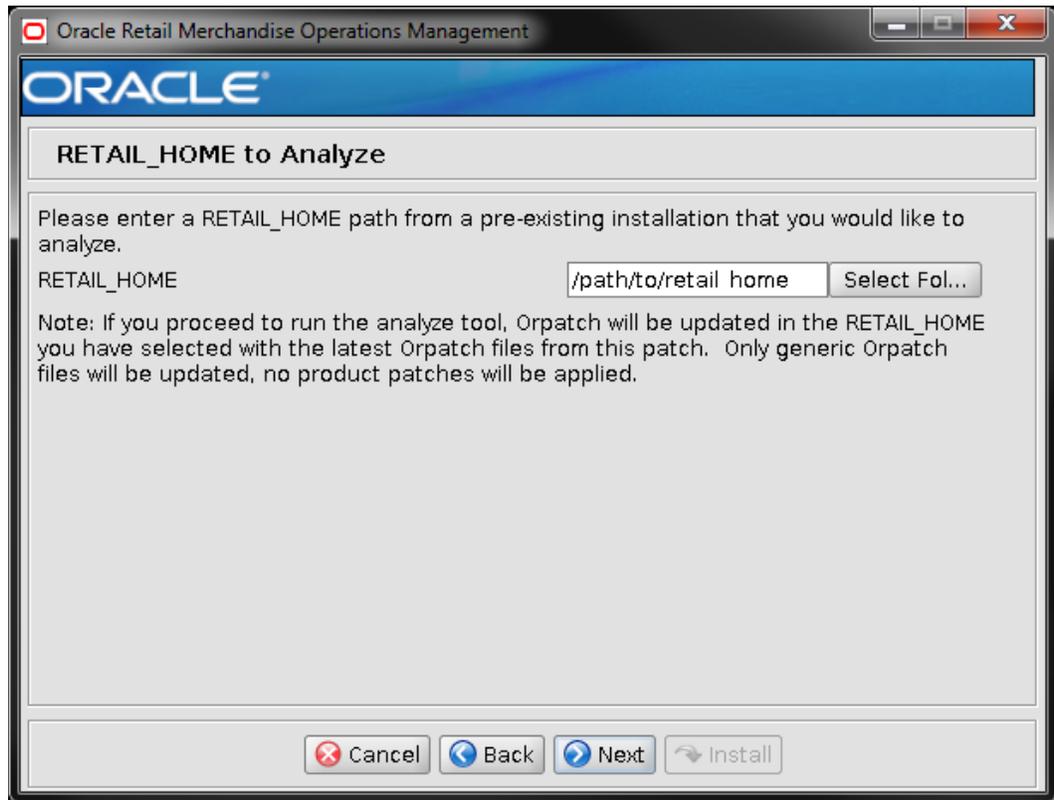
Variable	Description	Example
JAVA_HOME	Location of a Java 1.7_21 64Bit JDK.	JAVA_HOME= /u00/webadmin/java/jdk1.7.0_21 export JAVA_HOME
DISPLAY	Address and port of X server on desktop system of user running install. Optional when running the Analyze tool	DISPLAY=<IP address>:0.0 export DISPLAY

4. If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
5. Run the analyze.sh script to start the analyze tool.

Note: Below are the usage details for analyze.sh. The typical usage for GUI mode is no arguments.

```
./analyze.sh [text | silent]
```

Screen: RETAIL_HOME to Analyze



Field Title	RETAIL_HOME
Field Description	The pre-existing location where RMS (database, batch, and/or application) was installed along with the ORPATCH utility. This location should contain directories with your installed files as well as the "orpatch" directory.
Example	/path/to/retail_home
Note	The ORPatch files in this RETAIL_HOME may need to be updated in order to be able to run the analysis. The Analyze tool will take care of this automatically.

- After clicking "install", the Analyze tool will generate a report of the files that will be patched if you apply this patch to the selected RETAIL_HOME. A high level report can be found in the log file: STAGING_DIR/rms/installer/logs/rms-analyze.<timestamp>.log.

The detailed list of patch files can be found in RETAIL_HOME/orpatch/logs/detail_logs/analyze/details/

Appendix: Installer Silent Mode

In addition to the GUI and text interfaces of the installer, there is a silent mode that can be run. This mode is useful if you wish to run a repeat installation without retyping the settings you provided in the previous installation. It is also useful if you encounter errors in the middle of an installation and wish to continue.

The installer runs in two distinct phases. The first phase involves gathering settings from the user. At the end of the first phase, a properties file named `ant.install.properties` is created with the settings that were provided. Then the second phase begins, where this properties file is used to provide your settings for the installation.

To skip the first phase and re-use the `ant.install.properties` file from a previous run, follow these instructions:

1. Edit the `ant.install.properties` file and correct any invalid settings that may have caused the installer to fail in its previous run.
2. Look for duplicate properties in the `ant.install.properties` file. Some properties are set on multiple pages to ensure default values when a page is only displayed under certain conditions. For example, if there are two instances of `input.property.name`, remove all but the last one.
3. Run the installer again with the **silent** argument.

Example: `install.sh silent`

Appendix: URL Reference

This appendix provides URL reference information.

JDBC URL for a Database

Used by the Java application and by the installer to connect to the database.

Thick Client Syntax: jdbc:oracle:oci:@<sid>

<sid>: system identifier for the database

Example: jdbc:oracle:oci:@mysid

Thin Client Syntax: jdbc:oracle:thin:@<host>:<port>:<sid>

<host>: hostname of the database server

<port>: database listener port

<sid>: system identifier for the database

Example: jdbc:oracle:thin:@myhost:1521:mysid

Appendix: Common Installation Errors

This section provides some common errors encountered during installation of RMS.

RMS Installer unable to connect to the database

Symptoms

RMS installer throws error after clicking Next on the installer screen providing database credentials. This happens inspite of the fact that pre-requisite steps for database pre-install check showing status as SUCCESS as below.

```
*****
Final preinstall status
*****
Database Preinstall Check:    SUCCESS
```

Observe that install console also shows error message like below:

```
ORACLE_HOME=/u00/webadmin/product/wls_forms
ORACLE_INSTANCE= /u00/webadmin/config/domains/wls_forms/ ClassicDomain
/config/fmwconfig/components/FORMS/instances/forms1
DOMAIN_HOME= /u00/webadmin/config/domains/wls_forms/ClassicDomain
FORMSAPP_HOME
=/u00/webadmin/config/domains/wls_forms/FormsDomain/config/fmwconfig/serv
ers/WLS_FORMS/applications/formsapp_12.2.1/config
WLS_INSTANCE= forms1
ORACLE_SID=dvols64
JAVA_HOME=/u00/webadmin/product/jdk_java

Application environment check successful
```

```
./install.sh[77]: .[162]: call_child_preinstall_scripts[94]:
source_preinstall_profile[57]: .[19]: export:
2.0.4.v201112161009.jar:/u00/webadmin/product/wls_retail/modules/com.orac
le.jpa2support_1.0.0.0_2-
1.jar:/u00/webadmin/product/wls_retail/modules/eclipselink.jar:/u00/webad
min/product/wls_retail/patch_wls1036/profiles/default/sys_manifest_classp
ath/weblogic_patch.jar:/u00/webadmin/product/jdk_java/lib/tools.jar:...
```

Solution

Backup <Middleware_home>/wlserver_10.3/common/bin/commEnv.sh and update the file removing new line entry at below 3 entries as single entry.

Example:

From:

```
PRE_CLASSPATH="${MODULES_DIR}/javax.persistence_
2.0.4.v201112161009.jar:${MODULES_DIR}/com.oracle.jpa2support_1.0.0.0_
2-1.jar:${MODULES_DIR}/eclipselink.jar"
```

To:

```
PRE_CLASSPATH="{MODULES_DIR}/javax.persistence_2.0.4.v201112161009.jar:${MODULES_DIR}/com.oracle.jpa2support_1.0.0.0_2-1.jar:${MODULES_DIR}/eclipselink.jar"
```

Restart installer after making the above change.

Database Installer Hangs on Startup

Symptom

When the database schema installer is run, the following is written to the console and the installer hangs indefinitely:

```
Running pre-install checks  
Running tnsping to get listener port
```

Solution

The installer startup script is waiting for control to return from the **tnsping** command, but **tnsping** is hanging. Type Control+C to cancel the installer, and investigate and solve the problem that is causing the **tnsping <sid>** command to hang. This can be caused by duplicate database listeners running.

Warning: Could Not Find X Input Context

Symptom

The following text appears in the console window during execution of the installer in GUI mode:

```
Couldn't find X Input Context
```

Solution

This message is harmless and can be ignored.

Unresponsive Country and Currency Drop-Downs

Symptom

In GUI mode, when you click on the drop-down list selection for the primary country or currency, the list does not appear, and this message appears in the console window:

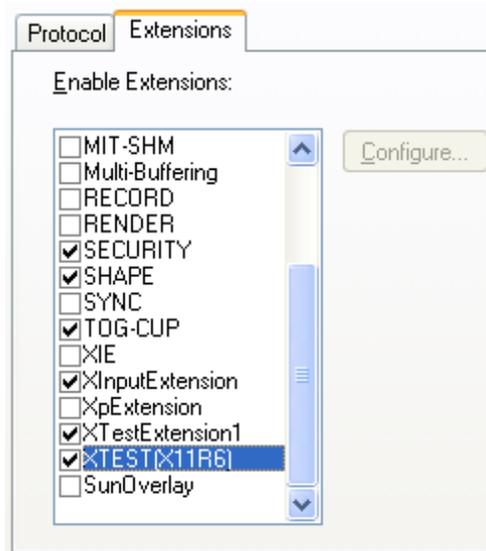
```
XTEST extension not installed on this X server: Error 0
```

Solution

To run the RMS installer in GUI mode you must have the XTEST extension enabled in your X server.

To Enabling XTEST in Exceed, do the following.

1. Open Xconfig to edit Exceed configuration.
2. Go to the X Server Protocol settings.
3. Click on the Extensions tab.
4. Make sure that the XTEST extension is selected, as shown.



5. Restart the X Server and re-run the RMS installer.

Could Not Execl Robot Child Process: Permission Denied

Symptom

When opening a drop-down list in GUI mode of the RMS installer, the installer freezes up and displays the following message in the console:

```
Couldn't execl robot child process: Permission denied
```

Solution

As the owner of the database ORACLE_HOME (i.e. **oracle**), grant execute permissions to the `awt_robot*` files under `$ORACLE_HOME/jdk/jre/lib`. The database schema installer uses `$ORACLE_HOME/jdk` for its `JAVA_HOME`.

Example (using SUN Solaris):

```
chmod a+x $ORACLE_HOME/jdk/jre/lib/sparc/awt_robot
chmod a+x $ORACLE_HOME/jdk/jre/lib/sparcv9/awt_robot
```

ConcurrentModificationException in Installer GUI

Symptom

In GUI mode, the errors tab shows the following error:

```
java.util.ConcurrentModificationException
    at
java.util.AbstractList$Itr.checkForComodification(AbstractList.java:448)
    at java.util.AbstractList$Itr.next(AbstractList.java:419)
... etc
```

Solution

You can ignore this error. It is related to third-party Java Swing code for rendering of the installer GUI and does not affect the retail product installation.

FRM-30064: Unable to Parse Statement Select While Compiling fm_ituda.fmb

Symptom

When running the application installer you get the following error:

```
FRM-30064: Unable to parse statement select vu.uda_desc, vu.uda_id from v_uda vu
where get_primary_lang = get_user_lang and vu.display_type = 'LV' union all
select nvl(t.translated_value, vu.uda_desc), vu.uda_id from tl_shadow t, v_uda
vu where get_primary_lang != get_user_lang and upper(vu.uda_desc) = t.key(+) and
get_user_lang = t.lang(+) and vu.display_type = 'LV' order by 1.
ORA-28112: failed to execute policy function
Record Group RG_UDA_LOV
Form: FM_ITUDALST
```

```
FRM-30085: Unable to adjust form for output.
```

```
Form not created
```

Solution

Disable the database filter policies by running `drop_filter_policy.sql`, run the application installer again and then run `add_filter_policy.sql`. Both files can be located with the database installer.

ORA-04031 (Unable to Allocate Memory) Error During Database Schema Installation

Symptom

When running the database schema installer you get the following error one or more times:

```
[ora:sqlplus] alter package
[ora:sqlplus] *
[ora:sqlplus] ERROR at line 1:
[ora:sqlplus] ORA-04031: unable to allocate 92120 bytes of shared memory ("shared
[ora:sqlplus] pool", "unknown object", "PL/SQL MPCODE", "BAMIMA: Bam Buffer")
```

Solution

There was not enough available memory in the shared pool on the database at the time of compilation. There are several choices to get past this error:

- Log into the database and attempt to recompile invalid objects in the database schema. Subsequent attempts to compile the same object(s) can be successful.
- Have a DBA increase the shared pool size on the database and re-run the installer from scratch on a new schema user.

X Error of Failed Request: BadWindow (Invalid Window Parameter)

Symptom

When compiling forms during the application installation you receive this error one or more times:

```
X Error of failed request: BadWindow (invalid Window parameter)
Major opcode of failed request: 18 (X_ChangeProperty)
Resource id in failed request: 0x1800002
Serial number of failed request: 432
Current serial number in output stream: 437
```

Solution

This error occurs when there are too many requests made to the X server. If this error occurs manually recompile the form.

For example,

```
frrmpcmp.sh userid=$UP module_type=form module=FORM_OR_MENU
```

RIB Errors

At random times, the RIB will get certain errors such as GETNXT(?,?,?,?,?) and/or ORA-21700 object does not exist or is marked for delete. This is very confusing because you may research and find that the object exists and is valid.

You must re-initialize the reference to reference an existing object as follows.

1. Bring down the RIB in question.
2. Run /RIB_INSTALL_DIR>/InstallAndCompileAllRibOracleObjects.sql.
3. Run another object validate script (ex: inv_obj_comp.sql) to make sure objects are valid. (Some may have deal locked in the end of the previous step.)
4. Bring up the RIB in question.

Error Connecting to Database URL

Symptom

After entering database credentials in the installer screens and hitting next, a message pops up with an error like this:

```
Error connecting to database URL <url> as user <user>
details...
```

The message prevents you from moving on to the next screen to continue the installation.

Solution

This error occurs when the installer fails to validate the user credentials you have entered on the screen. Make sure that you have entered the credentials properly. If you receive a message similar to this:

```
Error connecting to database URL <url> as user <user>
java.lang.Exception: UnsatisfiedLinkError encountered when using the Oracle
driver.
Please check that the library path is set up properly or switch to the JDBC thin
client.
```

It may mean that the installer is using the incorrect library path variables for the platform you are installing on. Open the file `<STAGING_DIR>/rms/installer/common/preinstall.sh` and make sure the variable `use32bit` is set to `True` if you are on a 32 bit platform, and `False` if you are on a 64 bit platform.

Multi-Threaded OCI Client Dumps Core after Reconnecting To Database

Symptom

If a multi-threaded Oracle client process that uses OCI to connect to a remote database loses connectivity with the database, it tries to reconnect and the client program continues to run. The program then dumps the core with the following stack trace, when Automatic Diagnostic Repository (ADR) is enabled.

```
skgfgio sdbgrfbibf_io_block_file dbgrfrbf_read_block_file dbgrmflrp_read_page
dbgrmblgmp_get_many_pages dbgrmmdrmd_read_relation_meta_data
dbgrmdora_open_record_access_full
dbgriporc_openrel_wcreate dbgrip_open_relation_access dbgrip_start_iterator
dbgrip_relation_iterator dbgruprac_read_adrctl...
```

Solution

Oracle Retail recommended you disable ADR (`diag_adr_enabled=OFF`, a `sqlnet.ora` parameter) while using multi-threaded OCI/OCCI application. `diag_adr_enabled` was introduced in Oracle 11g as a new method of tracing ADR. This will dump additional trace details.

Disabling '`diag_adr_enabled`' does not disturb any functionality. Therefore, it can safely be unset by doing `diag_adr_enabled=off` in `sqlnet.ora`. However, if you still want tracing, you can have following parameters/variables set in `sqlnet.ora`:

```
trace_level_server=16 -- for server side NET tracing
trace_level_client=16 -- for client side NET tracing
```

For additional information on how to set traditional tracing, see the My Oracle Support document, "SQL*Net, Net8, Oracle Net Services - Tracing and Logging at a Glance" (ID 219968.1).

Forms Installer Fails on HP-UX

Symptom

Errors occur during Forms installer screens when run on HP-UX. When you click Next on the installer screen "Data Source Details" you get an error message on the screen saying ""no ocijdbc11 in java.library.path" that prevents you from moving to the next screen.

Solution

This error message can be ignored. Verify that the data source details you entered are correct, and uncheck the box labeled **Test Data Source?** The installer screens will not attempt to validate the data source when you click **Next**. The installer will attempt to validate once again when installation starts, and the installer will fail if the credentials are incorrect.

FRM -93552: cannot connect to runtime process. Error when using RMS in an SSO environment

Symptom

When launching multiple applications in a SSO environment RMS forms can fail with:

FRM-93552: cannot connect to runtime process.

Solution

You need to change the default JSESSIONID cookie name for the forms process. There are two articles from Oracle Support that document this process:

- How to Change the Default JSESSIONID Cookie Name for Forms (Doc ID 1578506.1)
- How To Redeploy the Forms Application after Modification of Forms J2EE Application Deployment Descriptors (Doc ID 1063045.1)

RUNNING 002065_retail_ctx.sql Creating Context 'RETAIL_CTX' CREATE CONTEXT RETAIL_CTX USING retail_ctx_pkg * ERROR at line 1: ORA-00955: name is already used by an existing object

Symptom

When re-running the installer in the same database, the installer may fail with the below error:

```
RUNNING 002065_retail_ctx.sql Creating Context 'RETAIL_CTX' CREATE CONTEXT  
RETAIL_CTX USING retail_ctx_pkg * ERROR at line 1: ORA-00955: name is already used  
by an existing object
```

Solution

The error means that the context RETAIL_CTX already exists. Since the context is a sys object, even if you drop and re-create the RMS schema, the context will remain. This needs to be explicitly dropped from sys login with the below statement.

```
Sql>drop context RETAIL_CTX;
```

After this, the installation may be resumed.

FRM-30085: Unable to adjust form for output

Symptom

When compiling forms, compilation fails with the error FRM-30085: Unable to adjust form for output. The forms log has either nothing showing in the output as an error, or it fails with cannot load library unit PM_RETAIL_API_SQL (referenced by INTERNAL_VARIABLES).

PM_RETAIL_API could be a variety of different packages.

Solution

Run the below query as sys and compile the output until no rows are returned

```
SQL> set heading off SQL> select 'alter '||do.object_type||' ' || owner ||  
'.'||do.object_name||' compile;'  
from dba_objects do,diana_version$ dv where dv.obj#=do.object_id and do.timestamp  
<> to_char(dv.stime,'YYYY-MM-DD:HH24:MI:SS')  
/  
alter PACKAGE RMS141MOCK.CURRENCY_SQL compile;  
alter PACKAGE RMS141MOCK.PM_RETAIL_API_SQL compile;  
  
SQL> alter PACKAGE RMS141MOCK.CURRENCY_SQL compile;  
Package altered.  
  
SQL> alter PACKAGE RMS141MOCK. PM_RETAIL_API_SQL compile;  
Package altered.  
  
SQL> select 'alter '||do.object_type||' ' || owner || '.'||do.object_name||'  
compile;' from dba_objects do,diana_version$ dv
```

```
where dv.obj#=do.object_id 4 and do.timestamp <> to_char(dv.stime,'YYYY-MM-  
DD:HH24:MI:SS')  
/  
no rows selected  
  
SQL>
```

When no rows are returned, rerun the installer.

Appendix: Single Sign-On for WebLogic

Single Sign-On (SSO) is a term for the ability to sign onto multiple Web applications via a single user ID/Password. There are many implementations of SSO. Oracle provides an implementation with Oracle Access Manager.

Most, if not all, SSO technologies use a session cookie to hold encrypted data passed to each application. The SSO infrastructure has the responsibility to validate these cookies and, possibly, update this information. The user is directed to log on only if the cookie is not present or has become invalid. These session cookies are restricted to a single browser session and are never written to a file.

Another facet of SSO is how these technologies redirect a user's Web browser to various servlets. The SSO implementation determines when and where these redirects occur and what the final screen shown to the user is.

Most SSO implementations are performed in an application's infrastructure and not in the application logic itself. Applications that leverage infrastructure managed authentication (such as deployment specifying Basic or Form authentication) typically have little or no code changes when adapted to work in an SSO environment.

What Do I Need for Single Sign-On?

A Single Sign-On system involves the integration of several components, including Oracle Identity Management and Oracle Access Management. This includes the following components:

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle Access Manager (OAM) 11g Release 2 server and administrative console for implementing and configuring policies for single sign-on.
- A Policy Enforcement Agent such as Oracle Access Manager 11g R2 PS3Agent (WebGate), used to authenticate the user and create the Single Sign-On cookies.
- Oracle Directory Services Manager (ODSM) application in Oracle Identity Management (11.1.1.9), used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OAM system and registering HTTP servers.

Additional WebLogic managed servers will be needed to deploy the business applications leveraging the Single Sign-On technology.

Can Oracle Access Manager Work with Other SSO Implementations?

Yes, Oracle Access Manager has the ability to interoperate with many other SSO implementations, but some restrictions exist.

Oracle Single Sign-on Terms and Definitions

The following terms apply to single sign-on.

Authentication

Authentication is the process of establishing a user's identity. There are many types of authentication. The most common authentication process involves a user ID and password.

Dynamically Protected URLs

A Dynamically Protected URL is a URL whose implementing application is aware of the Oracle Access Manager environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic protection typically display a Login link to provide user authentication and gain greater access to the application's resources.

Oracle Identity Management (OIM) and Oracle Access Manager (OAM) for 11g

Oracle Identity Management (OIM) 11g includes Oracle Internet Directory and ODSM. Oracle Access Manager (OAM) 11g R2 should be used for SSO using WebGate. Oracle Forms 11g contains Oracle HTTP server and other Retail Applications will use Oracle WebTier11g for HTTP Server.

MOD_WEBLOGIC

mod_WebLogic operates as a module within the HTTP server that allows requests to be proxied from the OracleHTTP server to the Oracle WebLogic server.

Oracle Access Manager 11g Agent (WebGate)

Oracle WebGates are policy enforcement agents which reside with relying parties and delegate authentication and authorization tasks to OAM servers.

Oracle Internet Directory

Oracle Internet Directory (OID) is an LDAP-compliant directory service. It contains user ids, passwords, group membership, privileges, and other attributes for users who are authenticated using Oracle Access Manager.

Partner Application

A partner application is an application that delegates authentication to the Oracle Identity Management Infrastructure. One such partner application is the Oracle HTTP Server (OHS) supplied with Oracle Forms Server or WebTier11g Server if using other Retail Applications other than Oracle Forms Applications.

All partner applications must be registered with Oracle Access Manager (OAM) 11g. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Statically Protected URLs

A URL is considered to be Statically Protected when an Oracle HTTP server is configured to limit access to this URL to only SSO authenticated users. Any unauthenticated attempt to access a Statically Protected URL results in the display of a login page or an error page to the user.

Servlets, static HTML pages, and JSP pages may be statically protected.

What Single Sign-On is not

Single Sign-On is NOT a user ID/password mapping technology.

However, some applications can store and retrieve user IDs and passwords for non-SSO applications within an OID LDAP server. An example of this is the Oracle Forms Web Application framework, which maps Single Sign-On user IDs to a database logins on a per-application basis.

How Oracle Single Sign-On Works

Oracle Access Manager involves several different components. These are:

- The Oracle Access Manager (OAM) server, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle Access Manager Agent associated with the Web application, which verifies and controls browser redirection to the Oracle Access Manager server.
- If the Web application implements dynamic protection, then the Web application itself is involved with the OAM system.

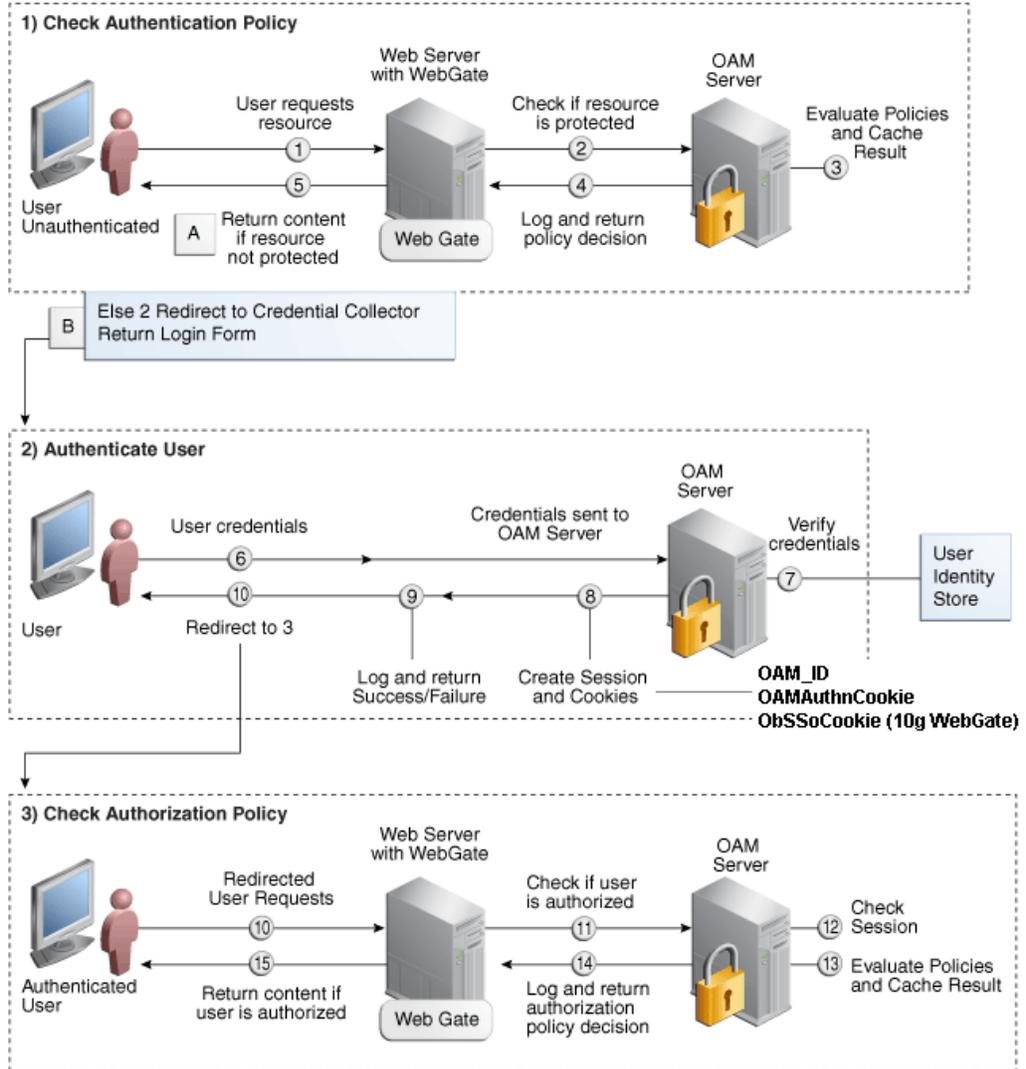
About SSO Login Processing with OAM Agents

1. The user requests a resource.
2. Webgate forwards the request to OAM for policy evaluation
3. OAM:
 - a. Checks for the existence of an SSO cookie.
 - b. Checks policies to determine if the resource is protected and if so, how?
4. OAM Server logs and returns the decision
5. Webgate responds as follows:
 - **Unprotected Resource:** Resource is served to the user
 - **Protected Resource:**
Resource is redirected to the credential collector.
The login form is served based on the authentication policy.
Authentication processing begins
6. User sends credentials
7. OAM verifies credentials
8. OAM starts the session and creates the following host-based cookies:
 - **One per partner:** OAMAuthnCookie set by 11g WebGates using authentication token received from the OAM Server after successful authentication.
Note: A valid cookie is required for a session.
 - **One for OAM Server:** OAM_ID
9. OAM logs Success or Failure.
10. Credential collector redirects to WebGate and authorization processing begins.
11. WebGate prompts OAM to look up policies, compare them to the user's identity, and determine the user's level of authorization.
12. OAM logs policy decision and checks the session cookie.
13. OAM Server evaluates authorization policies and cache the result.
14. OAM Server logs and returns decisions

15. WebGate responds as follows:

- If the authorization policy allows access, the desired content or applications are served to the user.
- If the authorization policy denies access, the user is redirected to another URL determined by the administrator.

SSO Login Processing with OAM Agents



Installation Overview

Installing an Oracle Retail supported Single Sign-On installation using OAM11g requires installation of the following:

1. Oracle Internet Directory (OID) LDAP server and the Oracle Directory Services Manager. They are typically installed using the Installer of Oracle Identity Management . The ODSM application can be used for user and realm management within OID.
2. Oracle Access Manager 11gR2 has to be installed and configured.
3. Additional midtier instances (such as Oracle Forms 11gr2) for Oracle Retail applications based on Oracle Forms technologies (such as RMS). These instances must be registered with the OAM installed in step 2.
4. Additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities must be registered with OAM installed in step 2.

Infrastructure Installation and Configuration

The Infrastructure installation for Oracle Access Manager (OAM) is dependent on the environment and requirements for its use. Deploying Oracle Access Manager (OAM) to be used in a test environment does not have the same availability requirements as for a production environment. Similarly, the Oracle Internet Directory (OID) LDAP server can be deployed in a variety of different configurations. See the *Oracle Identity Management Installation Guide*11g.

OID User Data

Oracle Internet Directory is an [LDAP v3](#) compliant directory server. It provides standards-based user definitions out of the box.

Customers with existing corporate LDAP implementations may need to synchronize user information between their existing LDAP directory servers and OID. OID supports standard LDIF file formats and provides a JNDI compliant set of Java classes as well. Moreover, OID provides additional synchronization and replication facilities to integrate with other corporate LDAP implementations.

Each user ID stored in OID has a specific record containing user specific information. For role-based access, groups of users can be defined and managed within OID. Applications can thus grant access based on group (role) membership saving administration time and providing a more secure implementation.

User Management

User Management consists of displaying, creating, updating or removing user information. There are many methods of managing an LDAP directory including LDIF scripts or Oracle Directory Services Manager (ODSM) available for OID11g.

ODSM

Oracle Directory Services Manager (ODSM) is a Web-based application used in OID11g is designed for both administrators and users which enables you to configure the structure of the directory, define objects in the directory, add and configure users, groups, and other entries. ODSM is the interface you use to manage entries, schema, security, adapters, extensions, and other directory features.

LDIF Scripts

Script based user management can be used to synchronize data between multiple LDAP servers. The standard format for these scripts is the LDAP Data Interchange Format (LDIF). OID supports LDIF script for importing and exporting user information. LDIF scripts may also be used for bulk user load operations.

User Data Synchronization

The user store for Oracle Access Manager resides within the Oracle Internet Directory (OID) LDAP server. Oracle Retail applications may require additional information attached to a user name for application-specific purposes and may be stored in an application-specific database. Currently, there are no Oracle Retail tools for synchronizing changes in OID stored information with application-specific user stores. Implementers should plan appropriate time and resources for this process. Oracle Retail strongly suggests that you configure any Oracle Retail application using an LDAP for its user store to point to the same OID server used with Oracle Access Manager.

Appendix: Single Sign-On Resource Access Descriptors

Oracle Forms applications such as RMS use database connections for authentication and authorization purposes. Oracle Single Sign-On, however, uses the Oracle Internet Directory (OID) user ID and password for this purpose. The Forms framework maps OID user IDs to database connections via information stored in Resource Access Descriptors (RADs). A user will have one RAD for each application accessed. RADs may be created by an administrator or by an LDIF script. Depending on the Oracle Internet Directory and/or the formsweb.cfg configuration, RADs may also be created by the user.

A user is prompted for the database connection information whenever formsweb.cfg file specifies ssoMode = true and createDynamicResources = true for an application and no valid RAD exists. RADs may become invalid when passwords have expired or have been changed.

RADs may be created by administrators or users via the Delegated Administration Services application.

Note: Users can create new RADs only if one or more RADs already exist.

RADs may be created and via LDIF scripts as well. See My Oracle Support document 244526.1.

Appendix: AIX Shared Library Bug Fix

The `env_rdbms.mk` file for Oracle 11g has Bug #2143531. This bug was not fixed because there is a workaround. For the workaround, the following changes in **bold/italic** need to be made to the `$ORACLE_HOME/rdbms/lib/env_rdbms.mk` file. Notice that changes are made in both the `BUILD_WITH_CONTEXT` and `BUILD_WITH_NO_CONTEXT` functions.

```

-----
BUILDLIB_WITH_CONTEXT=generate_export_list() \
{ \
/bin/nm -X32_64 -B -h -g "$$1" | grep -v ' U ' | awk '{print $$3}' | \
egrep -v '^\.|^TOC' | sort | uniq ; \
}; \
generate_import_list() { \
LIB_NAME=$$1; \
IMP_FILE=$$2; \
\
cat ${ORACLE_HOME}/rdbms/lib/xa.imp | head -1 | awk '{print $$0, "."}' >
${IMP_FILE}; \
/bin/nm -X32_64 -C -B -h -g ${LIB_NAME} | grep ' U ' | grep -v "::" | grep -v "("
| grep -v "\.cc" | awk '{print $$3}' | sed -e "s/\./g
" | grep -v "^_" >> ${IMP_FILE}; \
}; \
\
generate_import_list "$(OBJS)" $(SHARED_LIBNAME).imp; \
generate_export_list $(OBJS) > $(SHARED_LIBNAME).exp; \
$(LD) -bnoentry -bM:SRE -bE:$(SHARED_LIBNAME).exp -bI:$(SHARED_LIBNAME).imp \
-o $(SHARED_LIBNAME) $(OBJS) -L$(ORACLE_HOME)/lib -lc_r -lm $(LLIBCLNTSH)
$(MATHLIB)
-----
BUILDLIB_NO_CONTEXT=generate_export_list() \
{ \
/bin/nm -X32_64 -B -h -g "$$1" | grep -v ' U ' | awk '{print $$3}' | \
egrep -v '^\.|^TOC' | sort | uniq ; \
}; \
generate_import_list() { \
LIB_NAME=$$1; \
IMP_FILE=$$2; \
\
cat ${ORACLE_HOME}/rdbms/lib/xa.imp | head -1 | awk '{print $$0, "."}' >
${IMP_FILE}; \
/bin/nm -X32_64 -C -B -h -g ${LIB_NAME} | grep ' U ' | grep -v "::" | grep -v "("
| grep -v "\.cc" | awk '{print $$3}' | sed -e "s/\./g
" | grep -v "^_" >> ${IMP_FILE}; \
}; \
\
generate_import_list "$(OBJS)" $(SHARED_LIBNAME).imp; \
generate_export_list $(OBJS) > $(SHARED_LIBNAME).exp; \
$(LD) -bnoentry -bM:SRE -bE:$(SHARED_LIBNAME).exp -bI:$(SHARED_LIBNAME).imp \
-o $(SHARED_LIBNAME) $(OBJS) -L$(ORACLE_HOME)/lib -lc_r -lm $(LLIBCLNTSH)
$(MATHLIB)

```

Appendix: Inserting New Languages

As the RMS database installation does not support inserting new languages that have not already been installed, this section documents how to manually insert new languages as either primary or secondary languages. In this section <lang> represents the two or three-letter code for the language you wish to insert. This is the list of supported codes and the languages they represent:

- de - German
- es - Spanish
- el - Greek
- fr - French
- hu - Hungarian
- hr - Croatian
- it - Italian
- ja - Japanese
- ko - Korean
- nl - Dutch
- pl - Polish
- ptb - Brazilian Portuguese
- ru - Russian
- sv - Swedish
- tr - Turkish
- zhs - Simplified Chinese
- zht - Traditional Chinese

Insert Primary Language Data

To change the primary language data, complete the following steps.

Note: These scripts are only for customers who wish to have a primary language of one of the non-English supported languages. Once you run one of these primary scripts, you will not be able to revert back to English as your primary language. The scripts are AL32UTF8 encoded. We recommend installing them into a database that has been set to AL32UTF8.

1. Change directories to
STAGING_DIR/rms/installer/mom15/Cross_Pillar/languages
2. Set the sqlplus session so that the encoding component of the NLS_LANG is AL32UTF8. For example AMERICAN_AMERICA.AL32UTF8.
3. Log into sqlplus with the RMS schema and run the following command:

```
SQL> @rms_primary_<lang>.sql
```

4. Check the log file rms_primary_<lang>.log for any errors.

Note: Only one language can be set as the primary language for the system.

Appendix: Setting Up Password Stores with wallets/credential stores

As part of an application installation, administrators must set up password stores for user accounts using wallets/credential stores. Some password stores must be installed on the application database side. While the installer handles much of this process, the administrators must perform some additional steps.

Password stores for the application and application server user accounts must also be installed; however, the installer takes care of this entire process.

ORACLE Retail Merchandising applications now have 3 different types of password stores. They are database wallets, java wallets, and database credential stores. Background and how to administer them below are explained in this appendix

About Database Password Stores and Oracle Wallet

Oracle databases have allowed other users on the server to see passwords in case database connect strings (username/password@db) were passed to programs. In the past, users could navigate to `ps -ef |grep <username>` to see the password if the password was supplied in the command line when calling a program.

To make passwords more secure, Oracle Retail has implemented the Oracle Software Security Assurance (OSSA) program. Sensitive information such as user credentials now must be encrypted and stored in a secure location. This location is called password stores or wallets. These password stores are secure software containers that store the encrypted user credentials.

Users can retrieve the credentials using aliases that were set up when encrypting and storing the user credentials in the password store. For example, if `username/password@db` is entered in the command line argument and the alias is called `db_username`, the argument to a program is as follows:

```
sqlplus /@db_username
```

This would connect to the database as it did previously, but it would hide the password from any system user.

After this is configured, as in the example above, the application installation and the other relevant scripts are no longer needed to use embedded usernames and passwords. This reduces any security risks that may exist because usernames and passwords are no longer exposed.

When the installation starts, all the necessary user credentials are retrieved from the Oracle Wallet based on the alias name associated with the user credentials.

There are three different types of password stores. One type explain in the next section is for database connect strings used in program arguments (such as `sqlplus /@db_username`). The others are for Java application installation and application use.

Setting Up Password Stores for Database User Accounts

After the database is installed and the default database user accounts are set up, administrators must set up a password store using the Oracle wallet. This involves assigning an alias for the username and associated password for each database user account. The alias is used later during the application installation. This password store must be created on the system where the application server and database client are installed.

This section describes the steps you must take to set up a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, see the *Oracle Database Security Guide*.

Note: In this section, <wallet_location> is a placeholder text for illustration purposes. Before running the command, ensure that you specify the path to the location where you want to create and store the wallet.

To set up a password store for the database user accounts, perform the following steps:

1. Create a wallet using the following command:

```
mkstore -wrl <wallet_location> -create
```

After you run the command, a prompt appears. Enter a password for the Oracle Wallet in the prompt.

Note: The `mkstore` utility is included in the Oracle Database Client installation.

The wallet is created with the auto-login feature enabled. This feature enables the database client to access the wallet contents without using the password. For more information, refer to the *Oracle Database Advanced Security Administrator's Guide*.

2. Create the database connection credentials in the wallet using the following command:

```
mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>
```

After you run the command, a prompt appears. Enter the password associated with the database user account in the prompt.

3. Repeat Step 2 for all the database user accounts.
4. Update the `sqlnet.ora` file to include the following statements:

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = <wallet_location>)))
SQLNET.WALLET_OVERRIDE = TRUE
SSL_CLIENT_AUTHENTICATION = FALSE
```

5. Update the `tnsnames.ora` file to include the following entry for each alias name to be set up.

```
<alias-name> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = <host>) (PORT = <port>))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = <service>)
    )
  )
```

In the previous example, <alias-name>, <host>, <port>, and <service> are placeholder text for illustration purposes. Ensure that you replace these with the relevant values.

Setting up Wallets for Database User Accounts

The following examples show how to set up wallets for database user accounts for the following applications:

- [For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI](#)

For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI

To set up wallets for database user accounts, do the following.

1. Create a new directory called wallet under your folder structure.

```
cd /projects/rms15/dev/
mkdir .wallet
```

Note: The default permissions of the wallet allow only the owner to use it, ensuring the connection information is protected. If you want other users to be able to use the connection, you must adjust permissions appropriately to ensure only authorized users have access to the wallet.

2. Create a sqlnet.ora in the wallet directory with the following content.

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA =
(DIRECTORY = /projects/rms15/dev/.wallet)) )
SQLNET.WALLET_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```

Note: WALLET_LOCATION must be on line 1 in the file.

3. Setup a tnsnames.ora in the wallet directory. This tnsnames.ora includes the standard tnsnames.ora file. Then, add two custom tns_alias entries that are only for use with the wallet. For example, sqlplus /@dvols29_rms01user.

```
ifile = /u00/oracle/product/12.1.0.2/network/admin/tnsnames.ora
```

Examples for a NON pluggable db:

```
dvols29_rms01user =
(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
(host = xxxxxx.us.oracle.com) (Port = 1521)))
(CONNECT_DATA = (SID = <sid_name> (GLOBAL_NAME = <sid_name>)))
```

```
dvols29_rms01user.world =
(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
(host = xxxxxx.us.oracle.com) (Port = 1521)))
(CONNECT_DATA = (SID = <sid_name>) (GLOBAL_NAME = <sid_name>)))
```

Examples for a pluggable db:

```
dvols29_rms01user =
(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
(host = xxxxxx.us.oracle.com) (Port = 1521)))
(CONNECT_DATA = (SERVICE_NAME = <pluggable db name>)))
```

```
dvols29_rms01user.world =
(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
(host = xxxxxx.us.oracle.com) (Port = 1521)))
(CONNECT_DATA = (SERVICE_NAME = <pluggable db name>)))
```

Note: It is important to not just copy the tnsnames.ora file because it can quickly become out of date. The ifile clause (shown above) is key.

4. Create the wallet files. These are empty initially.
 - a. Ensure you are in the intended location.

```
$ pwd
/projects/rms15/dev/.wallet
```
 - b. Create the wallet files.

```
$ mkstore -wrl . -create
```
 - c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.
 - d. Enter the password again.

Two wallet files are created from the above command:

 - ewallet.p12
 - cwallet.sso
5. Create the wallet entry that associates the user name and password to the custom tns alias that was setup in the wallet's tnsnames.ora file.

```
mkstore -wrl . -createCredential <tns_alias> <username> <password>
```

Example: `mkstore -wrl . -createCredential dvols29_rms01user rms01user passwd`

6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.

```
$ export TNS_ADMIN=/projects/rms15/dev/.wallet /* This is very import to use
wallet to point at the alternate tnsnames.ora created in this example */
```

```
$ sqlplus /@dvols29_rms01user
```

```
SQL*Plus: Release 12
```

```
Connected to:
Oracle Database 12g
```

```
SQL> show user
USER is "rms01user"
```

Running batch programs or shell scripts would be similar:

```
Ex: dtesys /@dvols29_rms01user
script.sh /@dvols29_rms01user
```

Set the UP unix variable to help with some compiles :

```
export UP=/@dvols29_rms01user
for use in RMS batch compiles, and RMS, RWMS, and ARI forms compiles.
```

As shown in the example above, users can ensure that passwords remain invisible.

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

```
mkstore -wrl . -deleteCredential dvols29_rms01user
```

- Change the password for a credential on wallet

```
mkstore -wrl . -modifyCredential dvols29_rms01user rms01user passwd
```

- List the wallet credential entries

```
mkstore -wrl . -list
```

This command returns values such as the following.

```
oracle.security.client.connect_string1
oracle.security.client.user1
oracle.security.client.password1
```

- View the details of a wallet entry

```
mkstore -wrl . -viewEntry oracle.security.client.connect_string1
```

Returns the value of the entry:

```
dvols29_rms01user
mkstore -wrl . -viewEntry oracle.security.client.user1
```

Returns the value of the entry:

```
rms01user
```

```
mkstore -wrl . -viewEntry oracle.security.client.password1
```

Returns the value of the entry:

```
Passwd
```

Setting up RETL Wallets

RETL creates a wallet under \$RFX_HOME/etc/security, with the following files:

- cwallet.sso
- jazn-data.xml
- jps-config.xml
- README.txt

To set up RETL wallets, perform the following steps:

1. Set the following environment variables:
 - ORACLE_SID=<retaildb>
 - RFX_HOME=/u00/rfx/rfx-13
 - RFX_TMP=/u00/rfx/rfx-13/tmp
 - JAVA_HOME=/usr/jdk1.6.0_12.64bit
 - LD_LIBRARY_PATH=\$ORACLE_HOME
 - PATH=\$RFX_HOME/bin:\$JAVA_HOME/bin:\$PATH
2. Change directory to \$RFX_HOME/bin.
3. Run setup-security-credential.sh.
 - Enter 1 to add a new database credential.
 - Enter the dbuseralias. For example, retl_java_rms01user.
 - Enter the database user name. For example, rms01user.
 - Enter the database password.
 - Re-enter the database password.
 - Enter D to exit the setup script.

4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

For example, to configure RETLforRPAS, modify the following entries in `$RETAIL_HOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env`.

- The RETL_WALLET_ALIASES should point to the Java wallet entry:
 - `export RETL_WALLET_ALIASES="retl_java_rms01user"`
 - The ORACLE_WALLET_ALIASES should point to the Oracle network wallet entry:
 - `export ORACLE_WALLET_ALIASES="dvols29_rms01user"`
 - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIASES:
 - `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIASES}"`
5. To change a password later, run `setup-security-credential.sh`.
 - Enter 2 to update a database credential.
 - Select the credential to update.
 - Enter the database user to update or change.
 - Enter the password of the database user.
 - Re-enter the password.

For Java Applications (SIM, ReIM, RPM, RIB, AIP, Alloc, ReSA, RETL)

For Java applications, consider the following:

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.
- Passwords are not used to update entries in Java wallets. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application that has been installed, the wallet is located in `<WEBLOGIC_DOMAIN_HOME>/retail/<appname>/config` Example: `/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15/config`
- Application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.
- Scripts are located in `<WEBLOGIC_DOMAIN_HOME>/retail/<appname>/retail-public-security-api/bin` for administering wallet entries.
- Example:
 - `/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/retail-public-security-api/bin`
- In this directory is a script to help you update each alias entry without having to remember the wallet details. For example, if you set the RPM database alias to `rms01user`, you will find a script called `update-RMS01USER.sh`.

Note: These scripts are available only with applications installed by way of an installer.

- Two main scripts are related to this script in the folder for more generic wallet operations: `dump_credentials.sh` and `save_credential.sh`.
- If you have not installed the application yet, you can unzip the application zip file and view these scripts in `<app>/application/retail-public-security-api/bin`.

- Example:
- /u00/webadmin/reim15/application/retail-public-security-api/bin

update-<ALIAS>.sh

update-<ALIAS>.sh updates the wallet entry for this alias. You can use this script to change the user name and password for this alias. Because the application refers only to the alias, no changes are needed in application properties files.

Usage:

```
update-<username>.sh <myuser>
```

Example:

```
/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15/retail-public-
security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass in.
/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15/retail-public-
security-api/bin>
```

dump_credentials.sh

dump_credentials.sh is used to retrieve information from wallet. For each entry found in the wallet, the wallet partition, the alias, and the user name are displayed. Note that the password is not displayed. If the value of an entry is uncertain, run save_credential.sh to resave the entry with a known password.

```
dump_credentials.sh <wallet location>
```

Example:

```
dump_credentials.sh location:
/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15/config
```

```
Retail Public Security API Utility
```

```
=====
```

```
Below are the credentials found in the wallet at the
location/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15c
onfig
```

```
=====
```

```
Application level key partition name:reim15
User Name Alias:WLS-ALIAS User Name:weblogic
User Name Alias:RETAIL-ALIAS User Name:retail.user
User Name Alias:LDAP-ALIAS User Name:RETAIL.USER
User Name Alias:RMS-ALIAS User Name:rms15mock
User Name Alias:REIMBAT-ALIAS User Name:reimbat
```

save_credential.sh

save_credential.sh is used to update the information in wallet. If you are unsure about the information that is currently in the wallet, use dump_credentials.sh as indicated above.

```
save_credential.sh -a <alias> -u <user> -p <partition name> -l <path of the
wallet file location where credentials are stored>
```

Example:

```
/u00/webadmin/mock15_testing/reim15/application/retail-public-security-api/bin>
save_credential.sh -l wallet_test -a myalias -p mypartition -u myuser
```

```
=====
Retail Public Security API Utility
=====
```

```
Enter password:
Verify password:
```

Note: -p in the above command is for partition name. You must specify the proper partition name used in application code for each Java application.

save_credential.sh and dump_credentials.sh scripts are the same for all applications. If using save_credential.sh to add a wallet entry or to update a wallet entry, bounce the application/managed server so that your changes are visible to the application. Also, save a backup copy of your cwallet.sso file in a location outside of the deployment path, because redeployment or reinstallation of the application will wipe the wallet entries you made after installation of the application. To restore your wallet entries after a redeployment/reinstallation, copy the backed up cwallet.sso file over the cwallet.sso file. Then bounce the application/managed server.

Usage

```
=====
Retail Public Security API Utility
=====
usage: save_credential.sh -au[plh]
E.g. save_credential.sh -a rms-alias -u rms_user -p rib-rms -l ./
-a,--userNameAlias <arg>          alias for which the credentials
needs to be stored
-h,--help                          usage information
-l,--locationofWalletDir <arg>     location where the wallet file is
created.If not specified, it creates the wallet under secure-credential-wallet
directory which is already present under the retail-public-security-api/
directory.
-p,--appLevelKeyPartitionName <arg> application level key partition name
-u,--userName <arg>                username to be stored in secure
credential wallet for specified alias*
```

How does the Wallet Relate to the Application?

The ORACLE Retail Java applications have the wallet alias information you create in an <app-name>.properties file. Below is the reim.properties file. Note the database information and the user are presented as well. The property called `datasource.credential.alias=RMS-ALIAS` uses the ORACLE wallet with the argument of RMS-ALIAS at the `csm.wallet.path` and `csm.wallet.partition.name = reim14` to retrieve the password for application use.

Reim.properties code sample:

```
datasource.url=jdbc:oracle:thin:@xxxxxxx.us.oracle.com:1521:pkols07
datasource.schema.owner=rms15mock
datasource.credential.alias=RMS-ALIAS
# =====
# ossa related Configuration
#
# These settings are for ossa configuration to store credentials.
# =====

csm.wallet.path=/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15co
nfig
csm.wallet.partition.name=reim15
```

How does the Wallet Relate to Java Batch Program use?

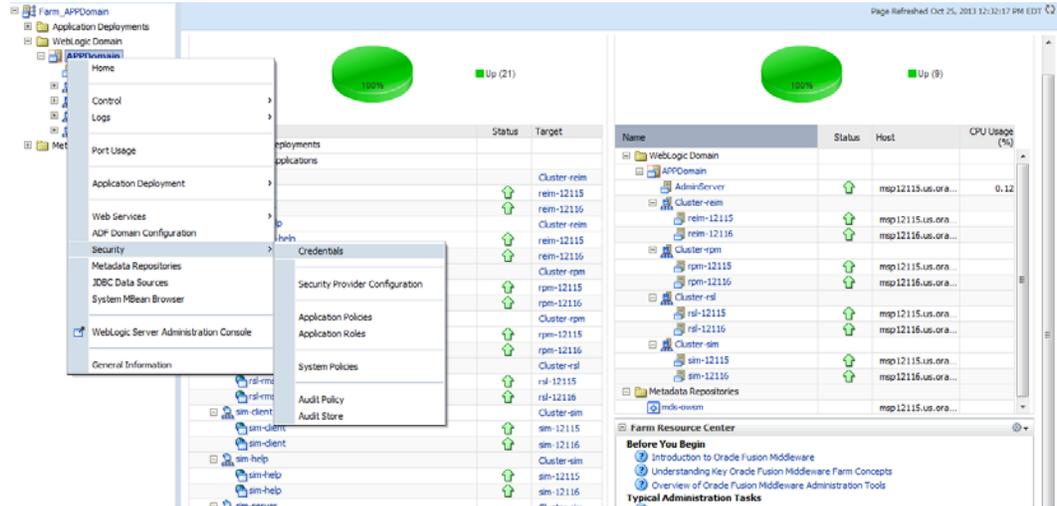
Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser RMS01APP, already on the database. To run a ReIM batch program the format would be: `reimbatchpgmname REIMBAT-ALIAS <other arguments as needed by the program in question>`

Database Credential Store Administration

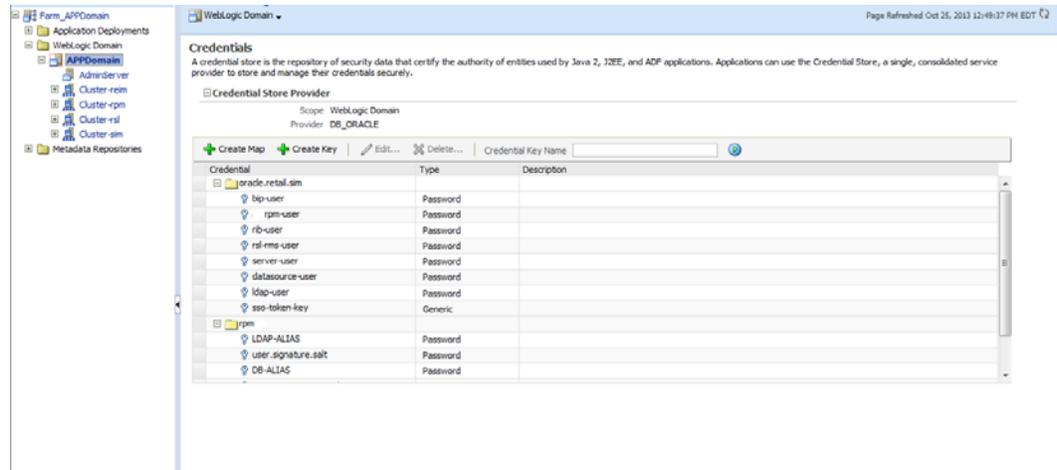
The following section describes a domain level database credential store. This is used in RPM login processing, SIM login processing, RWMS login processing, RESA login processing and Allocation login processing and policy information for application permission. Setting up the database credential store is addressed in the RPM, SIM, RESA, RWMS, and Alloc 15.0 install guides.

The following sections show an example of how to administer the password stores thru ORACLE Enterprise Manger Fusion Middleware Control, a later section will show how to do this thru WLST scripts.

1. The first step is to use your link to Oracle Enterprise Manager Fusion Middleware Control for the domain in question. Locate your domain on the left side of the screen and do a right mouse click on the domain and select **Security > Credentials**

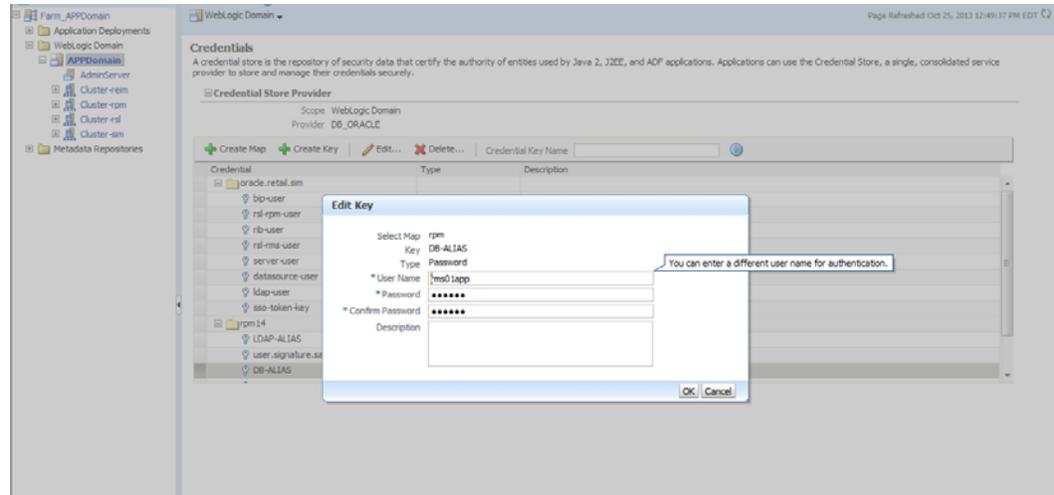


2. Click on Credentials and you will get a screen similar to the following. The following screen is expanded to make it make more sense. From here you can administer credentials.



The Create Map add above is to create a new map with keys under it. A map would usually be an application such as rpm15. The keys will usually represent alias to various users (database user, WebLogic user, LDAP user, etc). The application installer should add the maps so you should not often have to add a map.

Creation of the main keys for an application will also be built by the application installer. You will not be adding keys often as the installer puts the keys out and the keys talk to the application. You may be using EDIT on a key to see what user the key/alias points to and possibly change/reset its password. To edit a key/alias, highlight the key/alias in question and push the edit icon nearer the top of the page. You will then get a screen as follows:



The screen above shows the map (rpm) that came from the application installer, the key (DB-ALIAS) that came from the application installer (some of the keys/alias are selected by the person who did the application install, some are hard coded by the application installer in question), the type (in this case password), and the user name and password. This is where you would check to see that the user name is correct and reset the password if needed. REMEMBER, a change to an item like a database password WILL make you come into this and also change the password. Otherwise your application will NOT work correctly.

Managing Credentials with WSLT/OPSS Scripts

This procedure is optional as you can administer the credential store through the Oracle enterprise manager associated with the domain of your application install for ReIM, RPM, SIM, RESA, or Allocation.

An Oracle Platform Security Scripts (OPSS) script is a WLST script, in the context of the Oracle WebLogic Server. An online script is a script that requires a connection to a running server. Unless otherwise stated, scripts listed in this section are online scripts and operate on a database credential store. There are a few scripts that are offline, that is, they do not require a server to be running to operate.

Read-only scripts can be performed only by users in the following WebLogic groups: Monitor, Operator, Configurator, or Admin. Read-write scripts can be performed only by users in the following WebLogic groups: Admin or Configurator. All WLST scripts are available out-of-the-box with the installation of the Oracle WebLogic Server.

WLST scripts can be run in interactive mode or in script mode. In interactive mode, you enter the script at a command-line prompt and view the response immediately after. In

script mode, you write scripts in a text file (with a py file name extension) and run it without requiring input, much like the directives in a shell script.

The weakness with the WSLT/OPSS scripts is that you have to already know your map name and key name. In many cases, you do not know or remember that. The database credential store way through enterprise manager is a better way to find your map and key names easily when you do not already know them. A way in a command line mode to find the map name and alias is to run orapki. An example of orapki is as follows:

```
/u00/webadmin/product/wls_apps/oracle_common/bin> ./orapki wallet display -
wallet
/u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmw
config
```

(where the path above is the domain location of the wallet)

Output of orapki is below. This shows map name of rpm and each alias in the wallet:

Requested Certificates:

User Certificates:

Oracle Secret Store entries:

```
rpm@#3#@DB-ALIAS
rpm@#3#@LDAP-ALIAS
rpm@#3#@RETAIL.USER
rpm@#3#@user.signature.salt
rpm@#3#@user.signature.secretkey
rpm@#3#@WEBLOGIC-ALIAS
rpm@#3#@WLS-ALIAS
```

Trusted Certificates:

Subject: OU=Class 1 Public Primary Certification Authority,O=VeriSign\, Inc.,C=US

OPSS provides the following scripts on all supported platforms to administer credentials (all scripts are online, unless otherwise stated. You need the map name and the key name to run the scripts below

- listCred
- updateCred
- createCred
- deleteCred
- modifyBootStrapCredential
- addBootStrapCredential

listCred

The script `listCred` returns the list of attribute values of a credential in the credential store with given map name and key name. This script lists the data encapsulated in credentials of type password only.

Script Mode Syntax

```
listCred.py -map mapName -key keyName
```

Interactive Mode Syntax

```
listCred(map="mapName", key="keyName")
```

The meanings of the arguments (all required) are as follows:

- `map` specifies a map name (folder).
- `key` specifies a key name.

Examples of Use:

The following invocation returns all the information (such as user name, password, and description) in the credential with map name `myMap` and key name `myKey`:

```
listCred.py -map myMap -key myKey
```

The following example shows how to run this command and similar credential commands with WLS:

```
/u00/webadmin/product/wls_apps/oracle_common/common/bin>
sh wlst.sh
```

```
Initializing WebLogic Scripting Tool (WLS)...
```

```
Welcome to WebLogic Server Administration Scripting Shell
```

```
wls:/offline> connect('weblogic','password123','xxxxxx.us.oracle.com:17001')
Connecting to t3://xxxxxx.us.oracle.com:17001 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain
'APPDomain'.
```

```
wls:/APPDomain/serverConfig> listCred(map="rpm",key="DB-ALIAS")
Already in Domain Runtime Tree
```

```
[Name : rms01app, Description : null, expiry Date : null]
```

```
PASSWORD:retail
```

```
*The above means for map rpm15 in APPDomain, alias DB-ALIAS points to database
user rms01app with a password of retail
```

updateCred

The script `updateCred` modifies the type, user name, and password of a credential in the credential store with given map name and key name. This script updates the data encapsulated in credentials of type password only. Only the interactive mode is supported.

Interactive Mode Syntax

```
updateCred(map="mapName", key="keyName", user="userName", password="passW",
[desc="description"])
```

The meanings of the arguments (optional arguments are enclosed by square brackets) are as follows:

- `map` specifies a map name (folder) in the credential store.
- `key` specifies a key name.
- `user` specifies the credential user name.
- `password` specifies the credential password.
- `desc` specifies a string describing the credential.

Example of Use:

The following invocation updates the user name, password, and description of the password credential with map name `myMap` and key name `myKey`:

```
updateCred(map="myMap", key="myKey", user="myUsr", password="myPassw")
```

createCred

The script `createCred` creates a credential in the credential store with a given map name, key name, user name and password. This script can create a credential of type password only. Only the interactive mode is supported.

Interactive Mode Syntax

```
createCred(map="mapName", key="keyName", user="userName", password="passW",
[desc="description"])
```

The meanings of the arguments (optional arguments are enclosed by square brackets) are as follows:

- `map` specifies the map name (folder) of the credential.
- `key` specifies the key name of the credential.
- `user` specifies the credential user name.
- `password` specifies the credential password.
- `desc` specifies a string describing the credential.

Example of Use:

The following invocation creates a password credential with the specified data:

```
createCred(map="myMap", key="myKey", user="myUsr", password="myPassw")
```

deleteCred

The script `deleteCred` removes a credential with given map name and key name from the credential store.

Script Mode Syntax

```
deleteCred.py -map mapName -key keyName
```

Interactive Mode Syntax

```
deleteCred(map="mapName",key="keyName")
```

The meanings of the arguments (all required) are as follows:

- `map` specifies a map name (folder).
- `key` specifies a key name.

Example of Use:

The following invocation removes the credential with map name `myMap` and key name `myKey`:

```
deleteCred.py -map myMap -key myKey
```

modifyBootstrapCredential

The offline script `modifyBootstrapCredential` modifies the bootstrap credentials configured in the default `jps` context, and it is typically used in the following scenario: suppose that the policy and credential stores are LDAP-based, and the credentials to access the LDAP store (stored in the LDAP server) are changed. Then this script can be used to seed those changes into the bootstrap credential store.

This script is available in interactive mode only.

Interactive Mode Syntax

```
modifyBootstrapCredential( jpsConfigFile="pathName", username="usrName",
password="usrPass" )
```

The meanings of the arguments (all required) are as follows:

- `jpsConfigFile` specifies the location of the file `jps-config.xml` relative to the location where the script is run. Example location:
/u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig. Example location of the bootstrap wallet is
/u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig/bootstrap
- `username` specifies the distinguished name of the user in the LDAP store.
- `password` specifies the password of the user.

Example of Use:

Suppose that in the LDAP store, the password of the user with distinguished name `cn=orcladmin` has been changed to `welcome1`, and that the configuration file `jps-config.xml` is located in the current directory. Then the following invocation changes the password in the bootstrap credential store to `welcome1`:

```
modifyBootstrapCredential( jpsConfigFile=' ./jps-config.xml ',
username='cn=orcladmin', password='welcome1' )
```

Any output regarding the audit service can be disregarded.

addBootstrapCredential

The offline script `addBootstrapCredential` adds a password credential with given map, key, user name, and user password to the bootstrap credentials configured in the default `jps` context of a `jps` configuration file.

Classloaders contain a hierarchy with parent classloaders and child classloaders. The relationship between parent and child classloaders is analogous to the object relationship of super classes and subclasses. The bootstrap classloader is the root of the Java classloader hierarchy. The Java virtual machine (JVM) creates the bootstrap classloader, which loads the Java development kit (JDK) internal classes and `java.*` packages included in the JVM. (For example, the bootstrap classloader loads `java.lang.String`.)

This script is available in interactive mode only.

Interactive Mode Syntax

```
addBootstrapCredential( jpsConfigFile="pathName", map="mapName", key="keyName",
username="usrName", password="usrPass" )
```

The meanings of the arguments (all required) are as follows:

- `jpsConfigFile` specifies the location of the file `jps-config.xml` relative to the location where the script is run. Example location:
/u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig

- `map` specifies the map of the credential to add.
- `key` specifies the key of the credential to add.
- `username` specifies the name of the user in the credential to add.
- `password` specifies the password of the user in the credential to add.

Example of Use:

The following invocation adds a credential to the bootstrap credential store:

```
addBootStrapCredential(jpsConfigFile='./jps-config.xml', map='myMapName',  
key='myKeyName', username='myUser', password='myPass')
```

Quick Guide for Retail Password Stores (db wallet, java wallet, DB credential stores)

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (RETAIL_HOME)>/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile, execution	Installer	n/a	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile	Installer	n/a	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	n/a	<Db_Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rwms schema owner>	Compile forms, execute batch	Installer	n/a	Alias hard-coded by installer
RPM batch plsql and sqlldr	DB	<RPM batch install dir>/.wallet	n/a	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	RPM plsql and sqlldr batches
RWMS auto-login	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	<RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms15inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	n/a	Alias hard-coded by installer

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	DB credential store		Map=rpm or what you called the app at install time.	Many for app use					<weblogic domain home>/config/fmwc onfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file.
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm	<rpm batch user name> is the alias. Yes, here alias name = user name	<rpm batch user name>	App, batch use	Installer	RETAIL.USER	
	JAVA	<retail_home>/orpatch/config/javaapp_rpm							Each alias must be unique
			retail_installer	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	weblogic-alias	
			retail_installer	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			retail_installer	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			retail_installer	<LDAP-ALIAS>	cn=rpm.admin,cn=Users,dc=us,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALIAS	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name, ex: reim>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name, ex: reim>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name, ex: reim>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebser vice-alias	
			<installed app name, ex: reim>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			<installed app name, ex: reim>	<LDAP-ALIAS>	cn=REIM.A DMIN,cn=Users,dc=users,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALIAS	
	JAVA	<retail_home>/orpatch/conf/javaapp_reim							Each alias must be unique
			retail_installer	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			retail_installer	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			retail_installer	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebser vice-alias	
			retail_installer	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			retail_installer	<LDAP-ALIAS>	cn=REIM.A DMIN,cn=Users,dc=users,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALIAS	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RESA app	DB credential store		Map=resaor what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file. The bootstrap directory under this directory has bootstrap cwallet.sso file.
RESA app	JAVA	<weblogic domain home>/retail/<deployed resa app name>/config							Each alias must be unique
			<installed app name>	<resa weblogic user alias>	<resa weblogic user name>	App use	Installer	wlsalias	
			<installed app name>	<resa schema db user alias>	<rmsdb shema user name>	App use	Installer	Resadb-alias	
			<installed app name>	<resa schema user alias>	<rmsdb shema user name>>	App use	Installer	resa-alias	
	JAVA	<retail_home>/orpatch/config/javaapp_resa							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			retail_installer	<resa weblogic user alias>	<resa weblogic user name>	App use	Installer	wlsalias	
			retail_installer	<resa schema db user alias>	<rmsdb shema user name>	App use	Installer	Resadb-alias	
	JAVA	<retail_home>/orpatch/config/javaapp_rasm							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
Alloc app	DB credential store		Map=alloc or what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwc onfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file. The bootstrap directory under this directory has bootstrap cwallet.sso file.
Alloc app	JAVA	<weblogic domain home>/retail/config							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms schema user alias>	<rms schema user name>	App use	Installer	dsallocAlias	
			<installed app name>	<alloc batch user alias>	<SYSTEM_ADMINISTRATOR>	Batch use	Installer	alloc14	
	JAVA	<retail_home>/orpatch/config/javaapp_alloc							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			retail_installer	<rms schema user alias>	<rms schema user name>	App use	Installer	dsallocAlias	
			retail_installer	<alloc batch user alias>	<SYSTEM_ADMINISTRATOR>	Batch use	Installer	alloc14	
	JAVA	<retail_home>/orpatch/config/javaapp_rasm							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
SIM app	DB credential store		Map=oracle.retail.sim	Aliases required for SIM app use					<weblogic domain home>/config/fmwc onfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file.
	JAVA	<weblogic domain home>/retail/<deployed sim app name>/batch/resources/conf	oracle.retail.sim	<sim batch user alias>	<sim batch user name>	App use	Installer	BATCH-ALIAS	
	JAVA	<weblogic domain home>/retail/<deployed sim app name>/wireless/resources/conf	oracle.retail.sim	<sim wireless user alias>	<sim wireless user name>	App use	Installer	WIRELESS-ALIAS	
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application
RETL	DB	<RETL home>/wallet	n/a	<target application user alias>	<target application db userid>	App use	Manual	<db>_<user>	User may vary depending on RETL flow's target application
RIB	JAVA	<RIBHOME DIR>/deployment-home/conf/security							<app> is one of aip, rfm, rms, rpm, sim, rwms, tafr
JMS			jms<1-5>	<jms user alias> for jms<1-5>	<jms user name> for jms<1-5>	Integration use	Installer	jms-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
WebLogic			rib-<app>-app-server-instance	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integration use	Installer	weblogic-alias	
Admin GUI			rib-<app>#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integration use	Installer	admin-gui-alias	
Application			rib-<app>#user-alias	<app weblogic user alias>	<app weblogic user name>	Integration use	Installer	app-user-alias	Valid only for aip, rpm, sim
DB			rib-<app>#app-db-user-alias	<rib-app database schema user alias>	<rib-app database schema user name>	Integration use	Installer	db-user-alias	Valid only for rfm, rms, rwms, tafr
Error Hospital			rib-<app>#hosp-user-alias	<rib-app error hospital database schema user alias>	<rib-app error hospital database schema user name>	Integration use	Installer	hosp-user-alias	
RFI	Java	<RFI-HOME>/retail-financial-integration-solution/service-based-integration/conf/security							
			<installed app name>	rfiAppServerAdminServerUserAlias	<rfi weblogic user name>	App use	Installer	rfiAppServerAdminServerUserAlias	
			<installed app name>	rfiAdminUiUserAlias	<ORFI admin user>	App use	Installer	rfiAdminUiUserAlias	
			<installed app name>	rfiDataSourceUserAlias	<ORFI schema user name>	App use	Installer	rfiDataSourceUserAlias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	ebsDataSourceUserAlias	<EBS schema user name>	App use	Installer	ebsDataSourceUserAlias	
			<installed app name>	smtpMailFromAddressAlias	<From email address>	App use	Installer	smtpMailFromAddressAlias	

Appendix: Creating User Synonyms

Please refer to \$RETAIL_HOME/orpatch/utilities/create_synonyms_one_user.sql.

```

-----
-- Copyright (C) 2013,2014, Oracle and/or its affiliates. All rights reserved.
-----
-- This script creates synonyms in one schema (the synonym schema) to all objects
-- in another schema (the owning schema)
-- Arguments: synonym_schema owning_schema
set serveroutput on size unlimited
set escape on

declare
    synonym_schema          varchar2(30);
    owning_schema          varchar2(30);
    run_schema              varchar2(30);
    missing_object         varchar2(130);
    prefix1                 varchar2(128);
    prefix2                 varchar2(128);

    cursor c_get_missing_object (ownerschema in varchar2, synschema in varchar2) is
        (select object_name
         from dba_objects
         where owner = upper(ownerschema)
              and object_type in ('TABLE', 'VIEW', 'CLUSTER', 'FUNCTION', 'PACKAGE',
'PROCEDURE', 'SEQUENCE', 'TYPE')
              and object_name not like 'DBC_%'
              and object_name not like 'BIN$%'
         union
         select synonym_name from dba_synonyms
         where
         owner = ownerschema
         and table_name in ('ARI_INTERFACE_SQL', 'RMS_NOTIFICATION_REC') and
synonym_name in ('ARI_INTERFACE_SQL', 'RMS_NOTIFICATION_REC'))
        MINUS
        select object_name
         from dba_objects
         where owner = upper(synschema)
         order by 1;

begin
    synonym_schema := sys.dbms_assert.schema_name(upper('&1'));
    owning_schema := sys.dbms_assert.schema_name(upper('&2'));
    run_schema := sys.dbms_assert.schema_name('&_USER');

    IF synonym_schema <> run_schema THEN
        prefix1:=sys.dbms_assert.enquote_name(synonym_schema, FALSE) || '.';
    ELSE
        prefix1:='';
    END IF;

    IF owning_schema <> run_schema THEN
        prefix2:=sys.dbms_assert.enquote_name(owning_schema, FALSE) || '.';
    ELSE
        prefix2:='';
    END IF;

```

```
open c_get_missing_object(owning_schema,synonym_schema);
LOOP
  fetch c_get_missing_object into missing_object;
  --When at end of objects, exit
  if c_get_missing_object%NOTFOUND then
    exit;
  end if;

  missing_object:=sys.dbms_assert.enquote_name(missing_object, FALSE);

  BEGIN
    execute immediate 'CREATE SYNONYM '||prefix1||missing_object||' FOR
  '||prefix2||missing_object;
    dbms_output.put_line('Created synonym '||prefix1||missing_object||'
pointing to '||prefix2||missing_object);
  EXCEPTION
    WHEN OTHERS THEN
      dbms_output.put_line('Create synonym FAILED '||missing_object||'
'||SQLCODE||' - '||SQLERRM);
  END;
END LOOP;
close c_get_missing_object;
EXCEPTION
  WHEN OTHERS THEN
    raise;
end;
/
```

Appendix: Manual Forms Compilation

To manually recompile forms, please use the ORCompile utility

This is only possible after installer has been run and configured Oracle Retail Patch Assistant.

- Set RETAIL_HOME environment variable
- \$RETAIL_HOME/orpatch/bin/orcompile -a RMS -t FORMS

Usage:

```
orcompile -a <app> -t <type>
```

Potential Apps and Types:

ALLOC => DB-ALC,DB-RMS

REIM => DB

RMS => BATCH,DB,DB-ASYNC,DB-DEMO,FORMS

Appendix: Manual Batch Compilation

To manually recompile batch, please use the ORCompile utility.

This is only possible after installer has been run and configured Oracle Retail Patch Assistant.

- Set RETAIL_HOME environment variable
- \$RETAIL_HOME/orpatch/bin/orcompile -a RMS -t BATCH

Usage:

```
orcompile -a <app> -t <type>
```

Potential Apps and Types:

ALLOC => DB-ALC,DB-RMS

REIM => DB

RMS => BATCH,DB,DB-ASYNC,DB-DEMO,FORMS

Appendix: Installation Order

This section provides a guideline as to the order in which the Oracle Retail applications should be installed. If a retailer has chosen to use some, but not all, of the applications the order is still valid less the applications not being installed.

Note: The installation order is not meant to imply integration between products.

Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM)
2. Oracle Retail Sales Audit (ReSA)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Invoice Matching (ReIM)
7. Oracle Retail Price Management (RPM)
8. Oracle Retail Allocation
9. Oracle Retail Mobile Merchandising (ORMM)
10. Oracle Retail Xstore Office
11. Oracle Retail Xstore Point-of-Service, including Xstore Point-of-Service for Grocery, and including Xstore Mobile
12. Oracle Retail Xstore Environment
13. Oracle Retail EFTLink
14. Oracle Retail Store Inventory Management (SIM), including Mobile SIM
15. Oracle Retail Predictive Application Server (RPAS)
16. Oracle Retail Batch Script Architecture (BSA)
17. Oracle Retail Demand Forecasting (RDF)
18. Oracle Retail Category Management Planning and Optimization/Macro Space Optimization (CMPO/MSO)
19. Oracle Retail Replenishment Optimization (RO)
20. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
21. Oracle Retail Regular Price Optimization (RPO)
22. Oracle Retail Merchandise Financial Planning (MFP)
23. Oracle Retail Size Profile Optimization (SPO)
24. Oracle Retail Assortment Planning (AP)
25. Oracle Retail Item Planning (IP)
26. Oracle Retail Item Planning Configured for COE (IP COE)
27. Oracle Retail Advanced Inventory Planning (AIP)
28. Oracle Retail Integration Bus (RIB)
29. Oracle Retail Services Backbone (RSB)

-
30. Oracle Retail Financial Integration (ORFI)
 31. Oracle Retail Data Extractor for Merchandising
 32. Oracle Retail Clearance Optimization Engine (COE)
 33. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
 34. Oracle Retail Insights, including Retail Merchandising Insights (previously Retail Merchandising Analytics) and Retail Customer Insights (previously Retail Customer Analytics)