

Oracle® Retail Price Management

Installation Guide

Release 14.1.2

E68056-02

March 2016

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Amandeep Bhatti, Jyothsna Kamuni, Sravana Kumar

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	ix
Preface	xi
Audience	xi
Related Documents	xi
Customer Support	xi
Review Patch Documentation	xi
Improved Process for Oracle Retail Documentation Corrections	xii
Oracle Retail Documentation on the Oracle Technology Network	xii
Conventions	xii
1 Preinstallation Tasks	1
Implementation Capacity Planning	1
Check Supported Database Server Requirements	2
Check Supported Application Server Requirements	3
Check Single Sign-On Requirements	3
Check Supported Client PC and Web Browser Requirements	4
Check Oracle Retail Software Dependencies	4
Supported Oracle Retail Products	4
Supported Oracle Retail Integration Technologies	4
Check Third-Party Software Dependencies	4
UNIX User Account Privileges to Install the Software	5
2 RAC and Clustering	7
3 Database Installation Tasks	9
RPM Schema	9
4 Application Installation Tasks	11
Steps to Install and Configure ADF11g	11
Steps to Create the Domain with ADF Libraries and Enterprise Manager:	17
Update the WebLogic.policy:	29
Start the Node Manager	30
Start the AdminServer (admin console)	31
Start the Managed Server	32
Change the default (file based) Credential Store to use the Oracle Database	33
Create Required Schemas with RCU	33
Set up OPSS Schema Data source in WebLogic domain	41
Associate Policy Store to Database	46
Configure LDAP authentication Preinstallation Steps (Initial Login to RPM)	51
Create OID Authentication Provider	65
Verify and Set OID Authenticator	71
Expand the RPM Application Distribution	72
Clustered Installations – Preinstallation Steps	73

Run the RPM Application Installer	73
Resolving Errors Encountered During Application Installation	74
Clustered Installations – Post-Installation Steps.....	74
Review and/or Configure Oracle Single Sign-On.....	75
Create the RPM SSO provider in the RPMdomain:	75
Sign the RPM Client Configuration Jar File	76
Transaction Timeout.....	77
Backups Created by Installer.....	77
Test the RPM Application.....	77
RPM Batch Scripts.....	79
RPM Batch Scripts that call sqlplus (plsql batch)	80
Online Help.....	81
Adding a User to the RPM Application.....	81
5 Patching Procedures	83
Oracle Retail Patching Process	83
Supported Products and Technologies	83
Patch Concepts	84
Patching Utility Overview	85
Changes with 14.1	85
Patching Considerations	86
Patch Types.....	86
Incremental Patch Structure	86
Version Tracking.....	86
Apply all Patches with Installer or ORPatch.....	87
Environment Configuration	87
Retained Installation Files.....	87
Reloading Content	87
Java Hotfixes and Cumulative Patches	88
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.....	116
ORPatch Log Files.....	116
Restarting ORPatch.....	116
Manual DBManifest Updates.....	116
Manual Restart State File Updates	118
DISPLAY Settings When Compiling Forms.....	118
JAVA_HOME Setting	118
Patching Prior to First Install.....	118
Providing Metadata to Oracle Support.....	119
A Appendix: RPM Application Installer Screens.....	121
B Appendix: Common Installation Errors.....	157
Keystore errors when signing rpm_client_config.jar	157
Unreadable buttons in the Installer	157
Left menu buttons missing in RPM Client	157
Warning: Could not find X Input Context.....	158
Failed RPM Login	158
GUI screens fail to open when running Installer.....	158
C Appendix: URL Reference	159
JDBC URL for a Database	159
JNDI Provider URL for an Application	159
D Appendix: Setting Up Password Stores with wallets/credential stores.....	161
About Database Password Stores and Oracle Wallet	161
Setting Up Password Stores for Database User Accounts.....	161
Setting up Wallets for Database User Accounts	163
For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI	163
Setting up RETL Wallets	165
For Java Applications (SIM, ReIM, RPM, RIB, AIP, Alloc, ReSA, RETL).....	166
How does the Wallet Relate to the Application?	169
How does the Wallet Relate to Java Batch Program use?.....	169
Database Credential Store Administration.....	169
Managing Credentials with WSLT/OPSS Scripts	173
listCred	174
updateCred	175
createCred	175
deleteCred.....	175
modifyBootStrapCredential	176

addBootstrapCredential	177
Quick Guide for Retail Password Stores (db wallet, java wallet, DB credential stores)	179
E Appendix: Single Sign-On for WebLogic	189
What Do I Need for Single Sign-On?	189
Can Oracle Access Manager Work with Other SSO Implementations?	189
Oracle Single Sign-on Terms and Definitions	190
What Single Sign-On is not.....	191
How Oracle Single Sign-On Works	191
Installation Overview	193
User Management.....	193
F Appendix: Installation Order	195
Enterprise Installation Order.....	195

Send Us Your Comments

Oracle Retail Price Management, Installation Guide, Release 14.1.2

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 new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. 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

Related Documents

For more information, see the following documents in the Oracle Retail Price Management Release 14.1.2 documentation set:

- *Oracle Retail Price Management Release Notes*
- *Oracle Retail Merchandising Batch Schedule*
- *Oracle Retail Merchandising Data Model*

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, 14.1) or a later patch release (for example, 14.1.2). 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

Oracle Retail product documentation is 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. 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

RPM is a client-server application. Its client side code runs in a WebStart Java Virtual machine instance, while its server side code runs in the Oracle WebLogic Server and accesses an Oracle Database server.

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 scarcity
- 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.

Check Supported Database Server Requirements

General requirements for a database server running Oracle Retail Price Management include:

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 for x86-64 (Actual hardware or Oracle virtual machine).▪ Red Hat Enterprise Linux 6 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">▪ Patch 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 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 rdbms binary, apply patch 19623450. Then follow the instructions on Oracle Database Java Developer's Guide 12c Release 1 to change JDK to 1.7. The document 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 the Oracle Retail Price Management application include the following.

Supported on:	Versions Supported:
Application Server OS	<p>OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.7). Options are:</p> <ul style="list-style-type: none"> Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine) Red Hat Enterprise Linux 6 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)
Application Server	<p>Oracle Fusion Middleware 11g Release 1 (11.1.1.7) Components:</p> <ul style="list-style-type: none"> Oracle WebLogic Server 11g Release 1 (10.3.6) Repository Creation Utility (RCU 11.1.1.7) Oracle ADF 11g Release 1 (11.1.1.7) Oracle Identity Management 11g Release 1 (11.1.1.7) Note: Oracle Internet Directory (OID) is the supported LDAP directory for Oracle Retail products. For alternate LDAP directories, refer to Oracle WebLogic documentation set. Java: JDK 1.7+ 64 bit <p>IMPORTANT: If there is an existing WebLogic installation on the server, you must upgrade it to WebLogic 10.3.6. All middleware components associated with WebLogic server should be upgraded to 11.1.1.7.</p> <p>Optional (required for SSO)</p> <ul style="list-style-type: none"> Oracle WebTier 11g (11.1.1.7) Oracle Access Manager 11g Release 2 (11.1.2.2) Note: A separate WebLogic 10.3.6 installation is required for Oracle Access Manager 11g. Oracle Access Manager Agent (WebGate) 11g Release 2 (11.1.2.2)

Check Single Sign-On Requirements

If RPM will not be 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 11.1.1.7 has been installed along with the components listed in the above Application Server requirements section. . Verify the Oracle Access Manager Agent is registered with the Oracle Access Manager 11gR2 as a partner application.

Check Supported Client PC and Web Browser Requirements

Requirement	Version
Operating system	Windows 7 or 8
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 version 11 Mozilla Firefox 38+

Check Oracle Retail Software Dependencies

The database portion of the RMS 14.1.2 application must be installed prior to installing RPM.

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)/Oracle Retail Trade Management (RTM)/Oracle Retail Sales Audit (ReSA)	14.1.2
Oracle Retail Allocation	14.1.2
Oracle Retail Store Inventory Management (SIM)	14.1.2
Oracle Retail POS Suite	14.1.2

Supported Oracle Retail Integration Technologies

Requirement	Version
Oracle Retail Integration Bus (RIB)	14.1.2
Oracle Retail Service Backbone (RSB)	14.1.2

Check Third-Party Software Dependencies

Hibernate 4.1.0 must be downloaded and the hibernate4.jar file just be extracted. The RPM application installation procedure specifies how to install this file. The link to download jars is present in the readme.txt inside the hibernate4 folder for RPM software.

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.

RAC and Clustering

Oracle Retail Price Management has been validated to run in two configurations on Linux:

- Standalone WebLogic and Database installations
- Real Application Cluster Database and WebLogic Server 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.

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) E48838-08

Database Installation Tasks

RPM Schema

The RPM database tables are installed with the RMS database schema. RMS 14.1.2 is a prerequisite of the RPM 14.1.2 installation.

Note: The listed are RPM owned tables. Data on these tables indicates that a transaction is likely in progress within the system. Any transactions related to those tables needs to be completed. Any client owned data (on RPM_STAGE% tables) should be backed up by the client outside those tables. Any application owned data (payload data, workspace data – aka “WS”, and RPM_BULK_CC% related tables) should be removed by completing necessary transactions to do so (i.e., extract payload data and execute purge payloads batch; execute the purge bulk conflict check artifacts batch; etc). All transactional data needs to be fully processed and purged prior to 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_OVERRIDE
 - RPM_BULK_CC_PE_SEQUENCE
 - RPM_BULK_CC_PE_THREAD
 - RPM_BULK_CC_TASK
-
-

Application Installation Tasks

Before proceeding, you must install Oracle WebLogic Server 11g Release 1 (10.3.6) and patches listed in the Chapter 1 of this document. The Oracle Retail Price Management application is deployed to a WebLogic Managed server within the WebLogic installation. It is assumed Oracle Database has already been configured and loaded with the appropriate Oracle Retail Price Management schemas for your installation. ADF 11.1.1.7 should also be installed on the WebLogic installation.

Installing a separate domain is mandated. It can be called "RPMdomain" (or something similar) and will be used to install the managed servers. The ADF libraries should be extended to this domain and the Enterprise Manager should be deployed.

Steps to Install and Configure ADF11g

Follow the steps below to install ADF.

1. Download the ADF installation zip from disk1 and extract it to a stage location.
2. Set the environment variables below:

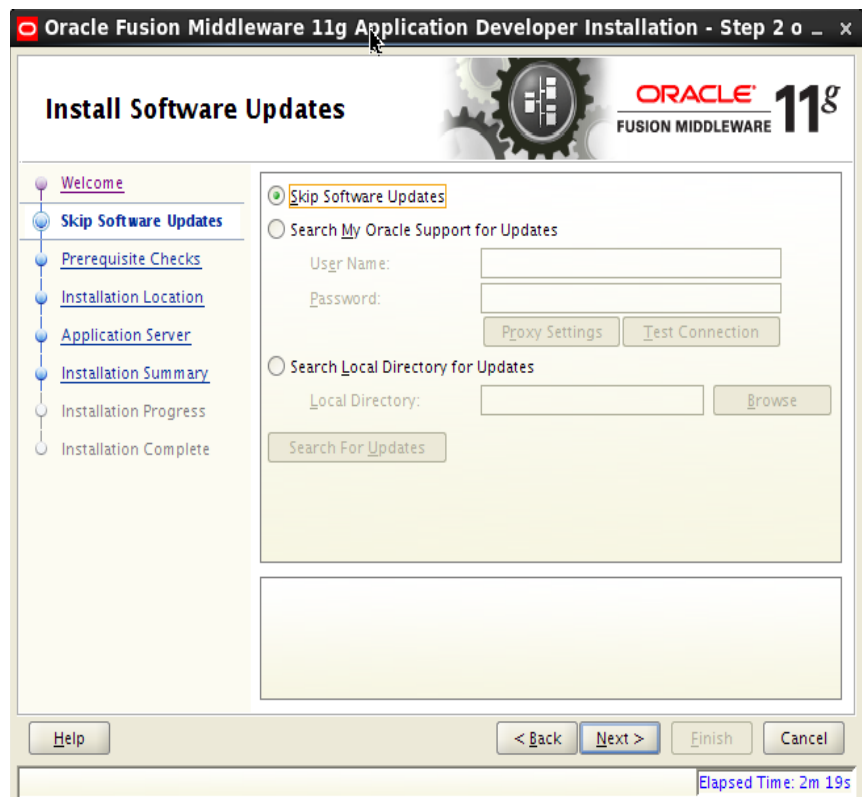
```
export JAVA_HOME=<location of JDK>
export PATH=$JAVA_HOME/bin:$PATH
```
3. Navigate to Disk1 directory in stage location and execute the installer command as below:

```
./runInstaller -jreLoc <JAVA_HOME>
```

4. The following Welcome screen is displayed. Click **Next**.



5. Select **Skip Software Updates** and click **Next**.



6. Click Next.



7. Enter the details as below:

Oracle Middleware Home = <This should be the Middleware Home location where Weblogic has been installed>. For example: /u00/webadmin/product/wls_retail

Oracle Home Directory = <leave this as default>. Eg: oracle_common

Click Next.

Oracle Fusion Middleware 11g Application Developer Installation - Step 4 o _ x

Specify Installation Location

[Welcome](#)
[Skip Software Updates](#)
[Prerequisite Checks](#)
[Installation Location](#)
[Application Server](#)
[Installation Summary](#)
[Installation Progress](#)
[Installation Complete](#)

Oracle **M**iddleware Home: lmin/product/fmw/ws_app

Oracle Home Directory: oracle_common

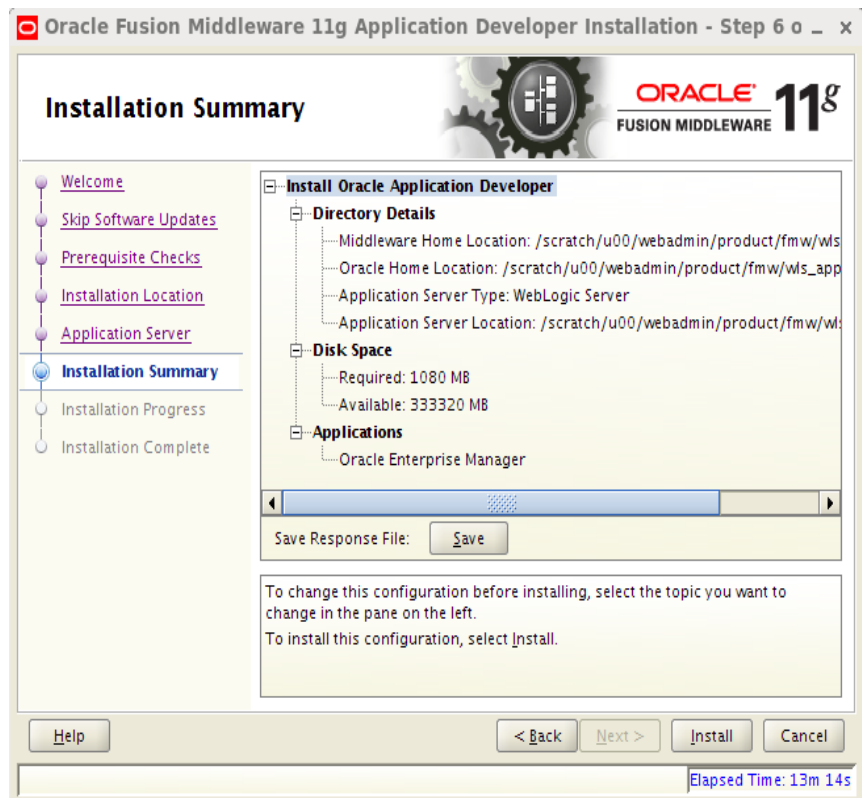
The Oracle home will be created under the Oracle Middleware Home location with the given name.

Elapsed Time: 1m 49s

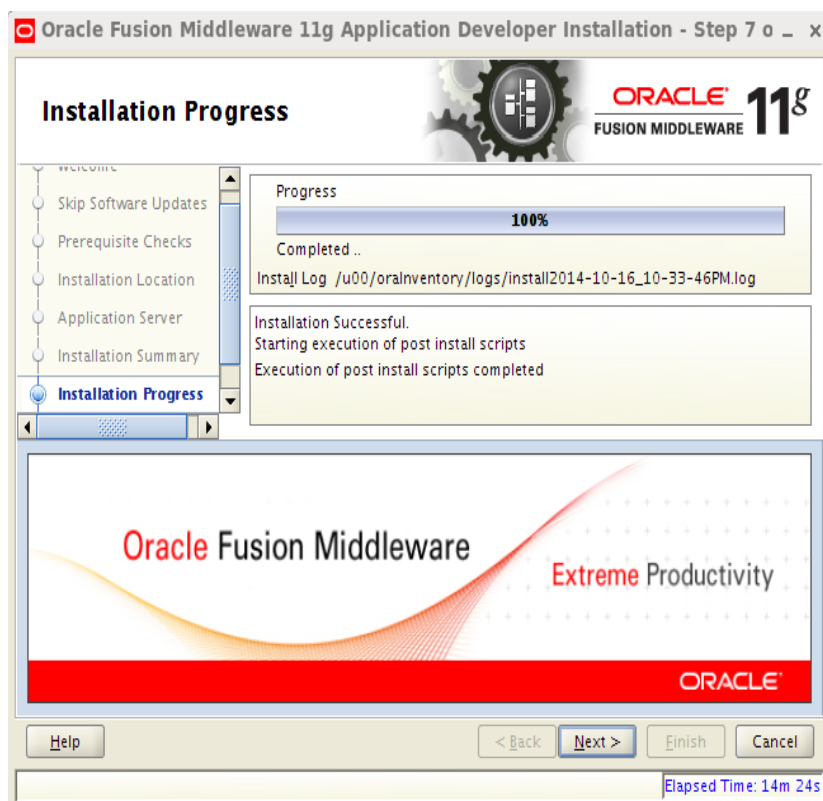
8. Select **WebLogic Server** and click **Next**.



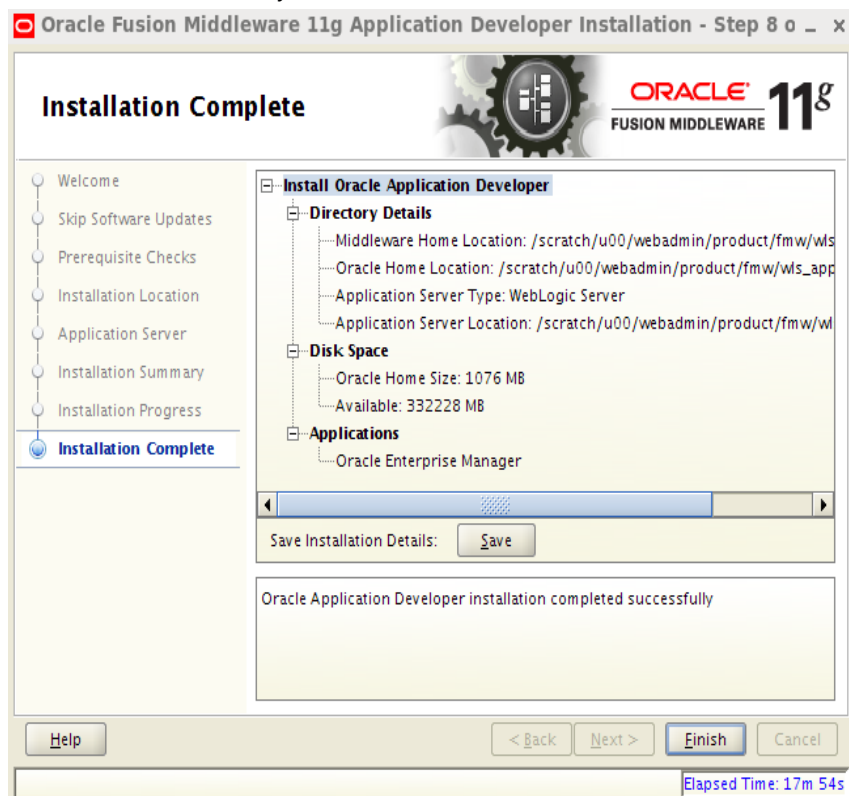
9. Click **Install**.



10. Click Next.



11. Click Save to save your installation details and click Finish.

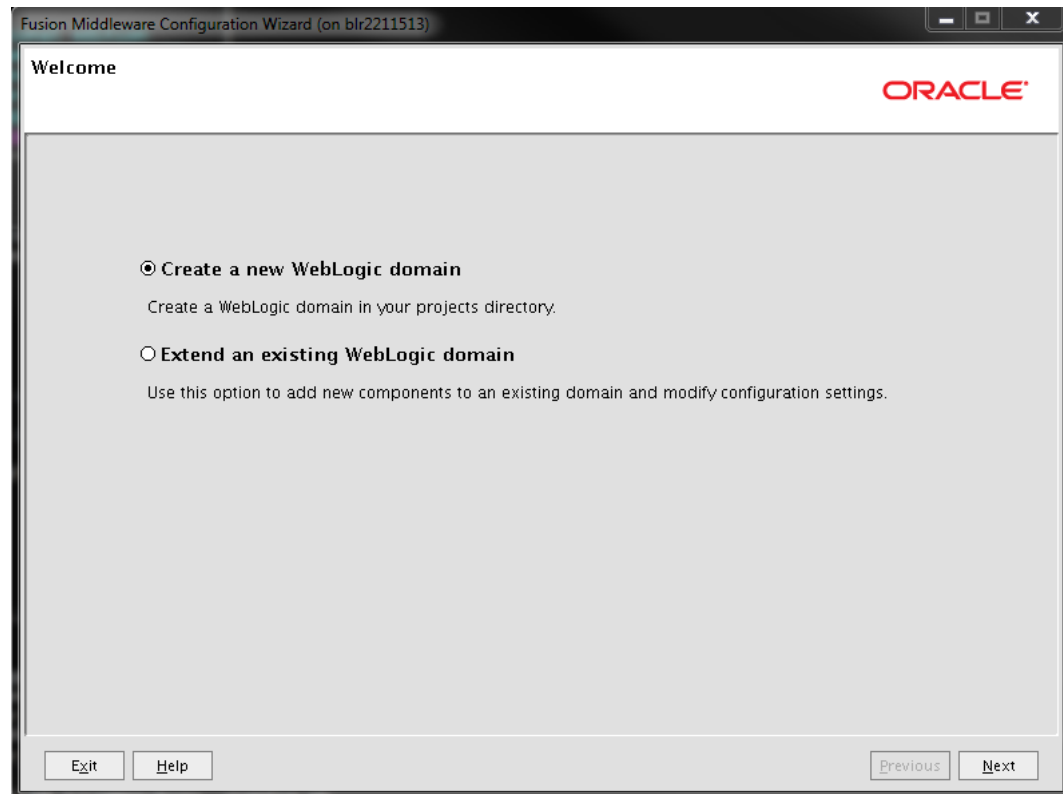


Steps to Create the Domain with ADF Libraries and Enterprise Manager:

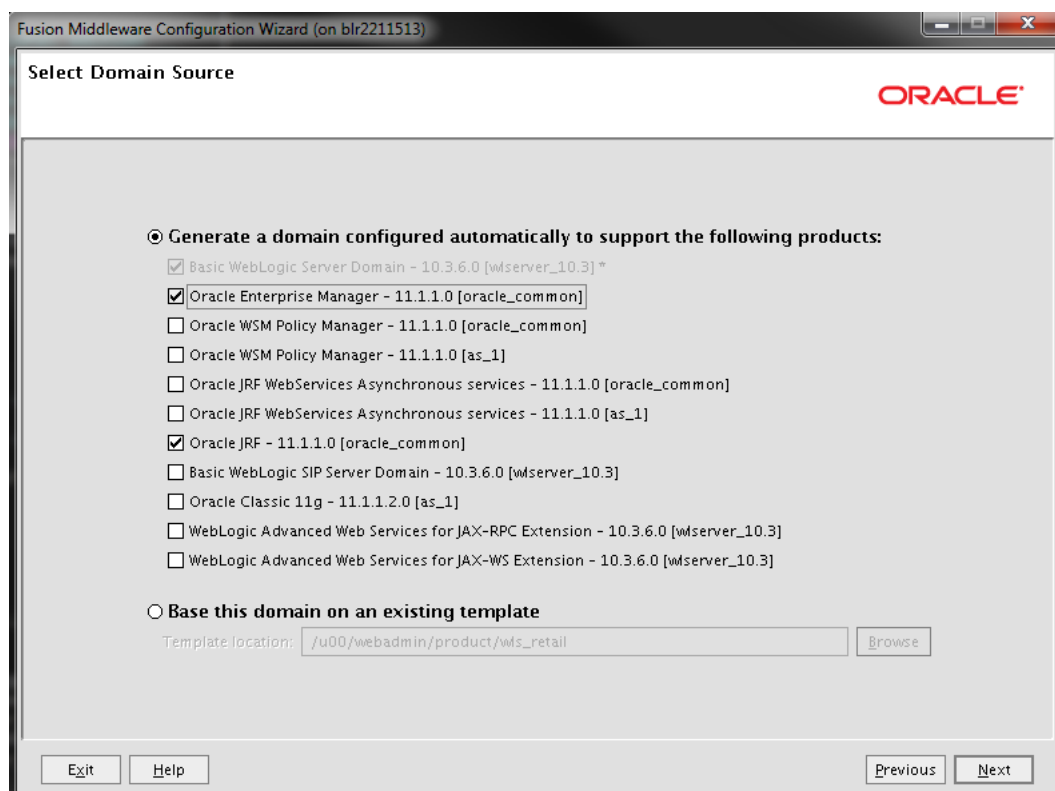
1. Set the required environment variables

```
export JAVA_HOME=<JDK_HOME>
export PATH=$JAVA_HOME/bin:$PATH
export WLS_HOME=<This should be the Weblogic Home location>
(Example:/u00/webadmin/product/wls_retail)
export
ORACLE_HOME=<WLS_HOME>/oracle_common(Example:/u00/webadmin/product/wls_retail/
oracle_common)
```

2. Change directories to \$WLS_HOME/wlserver_10.3/common/bin and run the config.sh scripts to create the new RPMdomain with Enterprise Manager. The following screen is displayed. Select the default and Click **Next**.



3. Select the Oracle JRF and the Oracle Enterprise Manager. Click **Next**.



4. Change the Domain name from the default. For example, RPMdomain. Click **Next**.

The screenshot shows a window titled "Fusion Middleware Configuration Wizard (on blr2211513)". The main heading is "Specify Domain Name and Location" with the Oracle logo in the top right. Below the heading is a large grey area with the text "Enter the name and location for the domain and its applications:". There are three input fields: "Domain name:" with the text "RPMdomain", "Domain location:" with the text "/u00/webadmin/product/wms_retail/user_projects/domains", and "Application location:" with the text "/u00/webadmin/product/wms_retail/user_projects/applications". Each of the last two fields has a "Browse" button to its right. At the bottom of the window are four buttons: "Exit", "Help", "Previous", and "Next".

Fusion Middleware Configuration Wizard (on blr2211513)

Specify Domain Name and Location **ORACLE**

Enter the name and location for the domain and its applications:

Domain name: RPMdomain

Domain location: /u00/webadmin/product/wms_retail/user_projects/domains Browse

Application location: /u00/webadmin/product/wms_retail/user_projects/applications Browse

Exit Help Previous Next

5. Enter 'User password' value and 'Confirm user password' value (same as user password).

User password=<password>

Confirm user password=<password>

Click **Next**.

Fusion Middleware Configuration Wizard (on blr2211513)

Configure Administrator User Name and Password ORACLE

[Discard Changes](#)

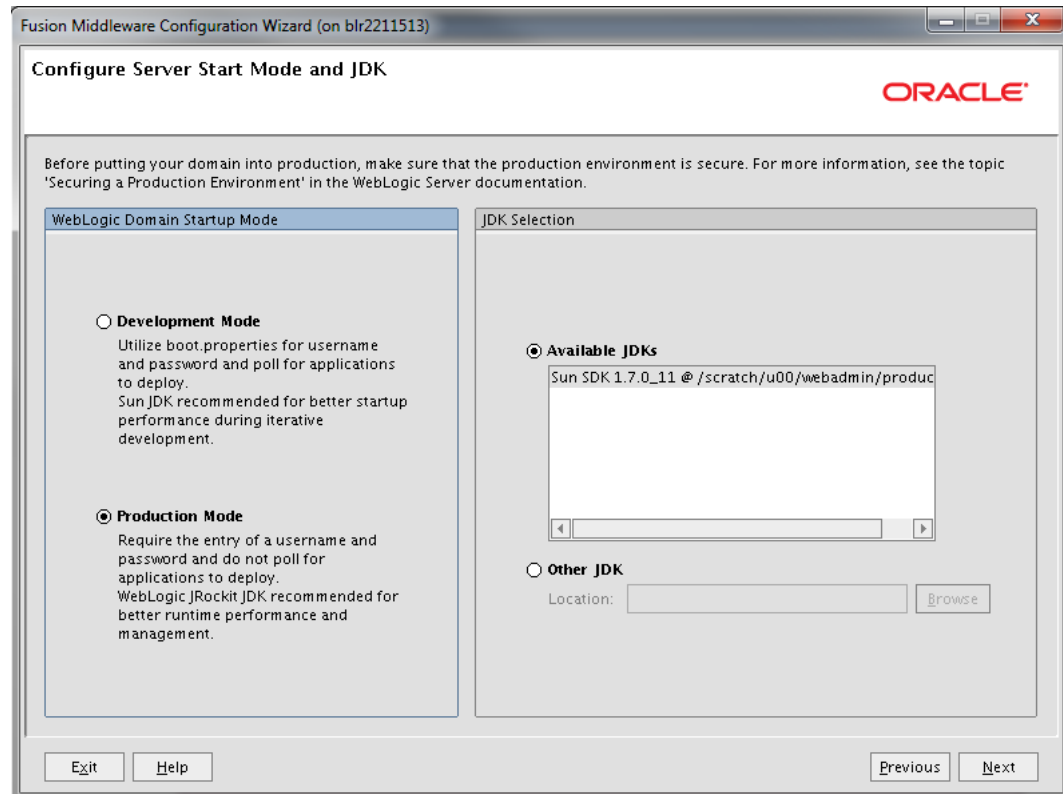
*Name:

*User password:

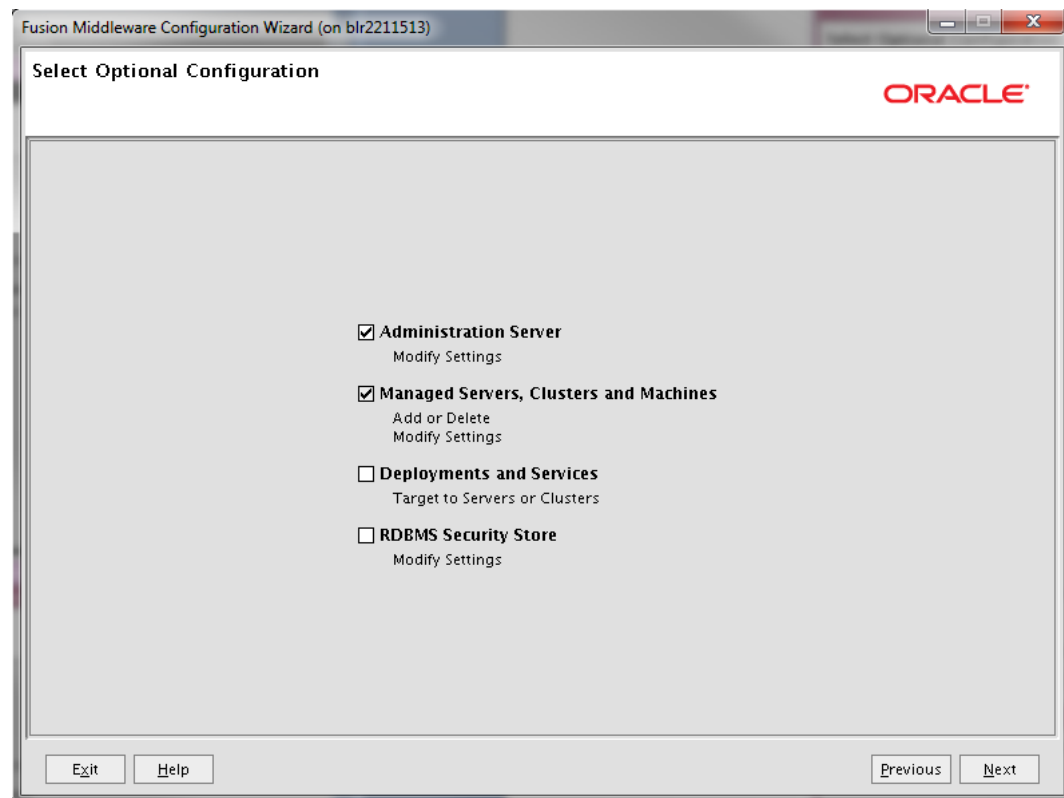
*Confirm user password:

Description:

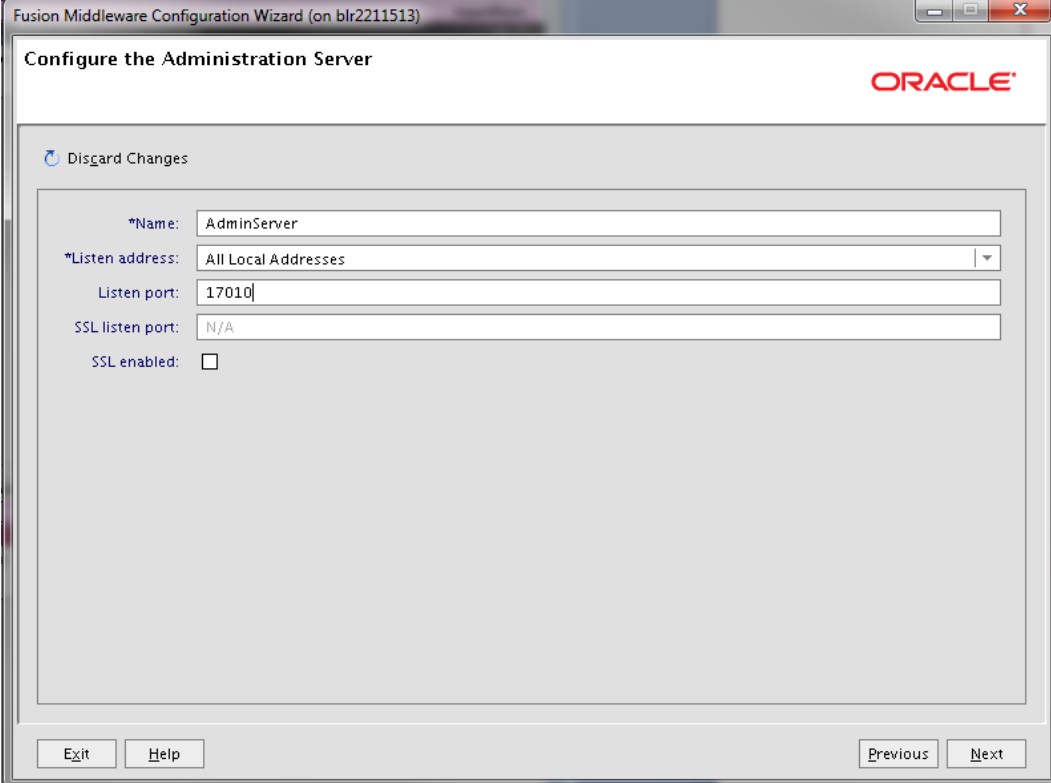
6. Select Production Mode for WebLogic domain Startup Mode. Click **Next**.



7. Select Administration Server as well as Managed Servers, Clusters and Machines. Click **Next**.



8. Enter the Listen port and click **Next**.
 - Listen port: 17010 (This port must be an open port on the server)



The screenshot shows the 'Fusion Middleware Configuration Wizard (on blr2211513)' window. The title bar includes standard Windows window controls. The main window has a title 'Configure the Administration Server' and the Oracle logo in the top right corner. Below the title bar, there is a 'Discard Changes' button with a circular arrow icon. The main configuration area contains several fields: '*Name:' with the value 'AdminServer', '*Listen address:' with a dropdown menu showing 'All Local Addresses', 'Listen port:' with the value '17010', 'SSL listen port:' with the value 'N/A', and 'SSL enabled:' with an unchecked checkbox. At the bottom of the window, there are four buttons: 'Exit', 'Help', 'Previous', and 'Next'.

Fusion Middleware Configuration Wizard (on blr2211513)

Configure the Administration Server

ORACLE

Discard Changes

*Name: AdminServer

*Listen address: All Local Addresses

Listen port: 17010

SSL listen port: N/A

SSL enabled: ☐

Exit Help Previous Next

9. Click Add and provide Name and Listen Port for the managed server.
 - Name: rpm-server (This is your managed server name)
 - Listen port: 17011 (This port must be an open port on the server)Click Next.

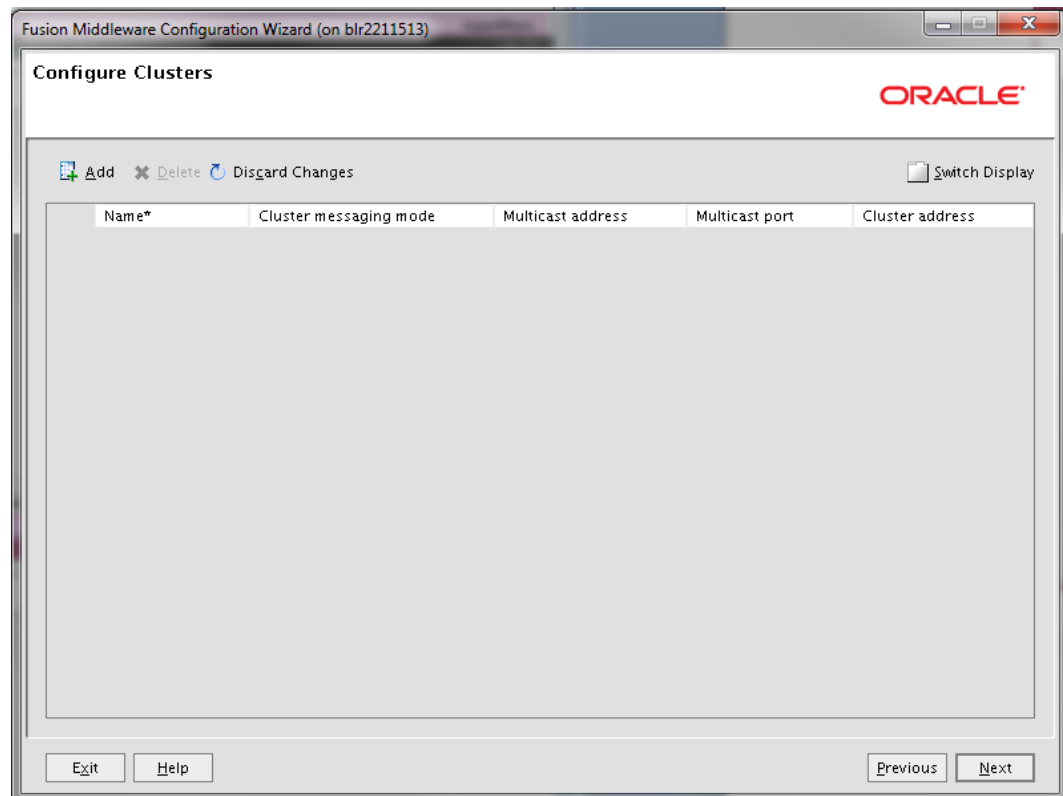
Fusion Middleware Configuration Wizard (on blr2211513)

Configure Managed Servers ORACLE

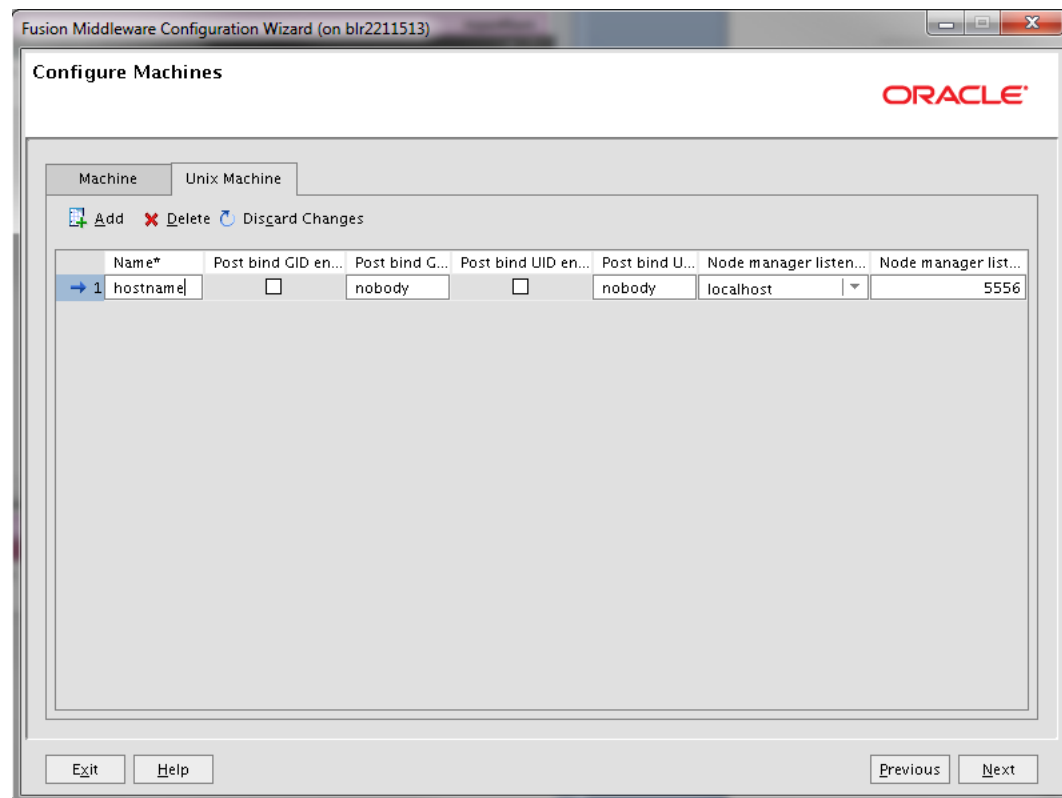
Add Delete Discard Changes Switch Display

	Name*	Listen address*	Listen port	SSL listen port	SSL enabled
→ 1	rpm-server	All Local Addresses	17011	N/A	<input type="checkbox"/>

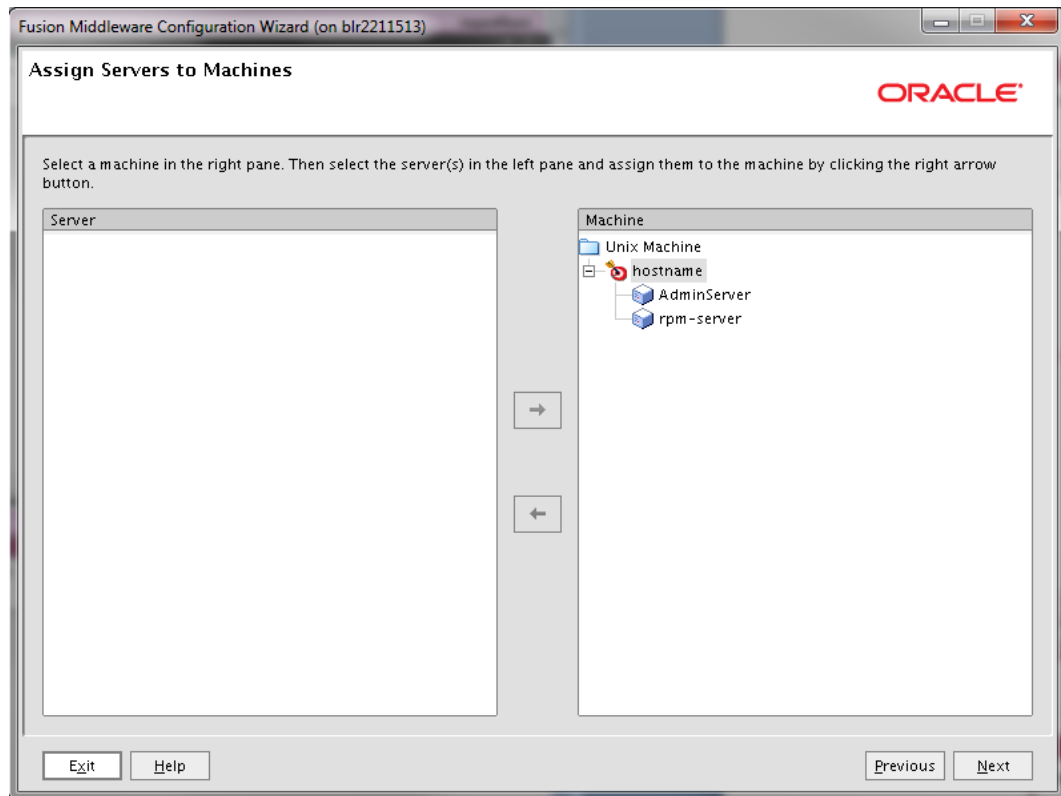
Exit Help Previous Next

10. Click Next.

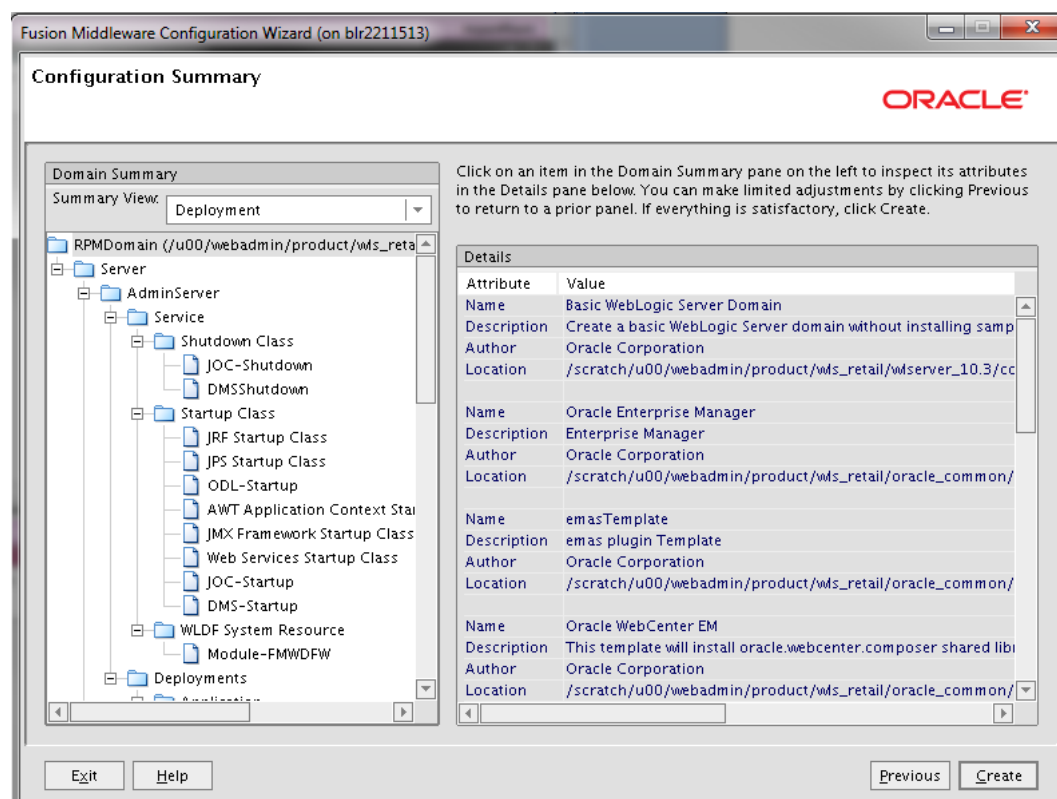
11. In 'Unix Machine' tab, Click Add and provide Name and an open port for Node manager listen port as shown below.
- Name: <hostname> (This can be any name or usually your hostname)
 - Listen port: 5556 (This port must be an open port on the server)



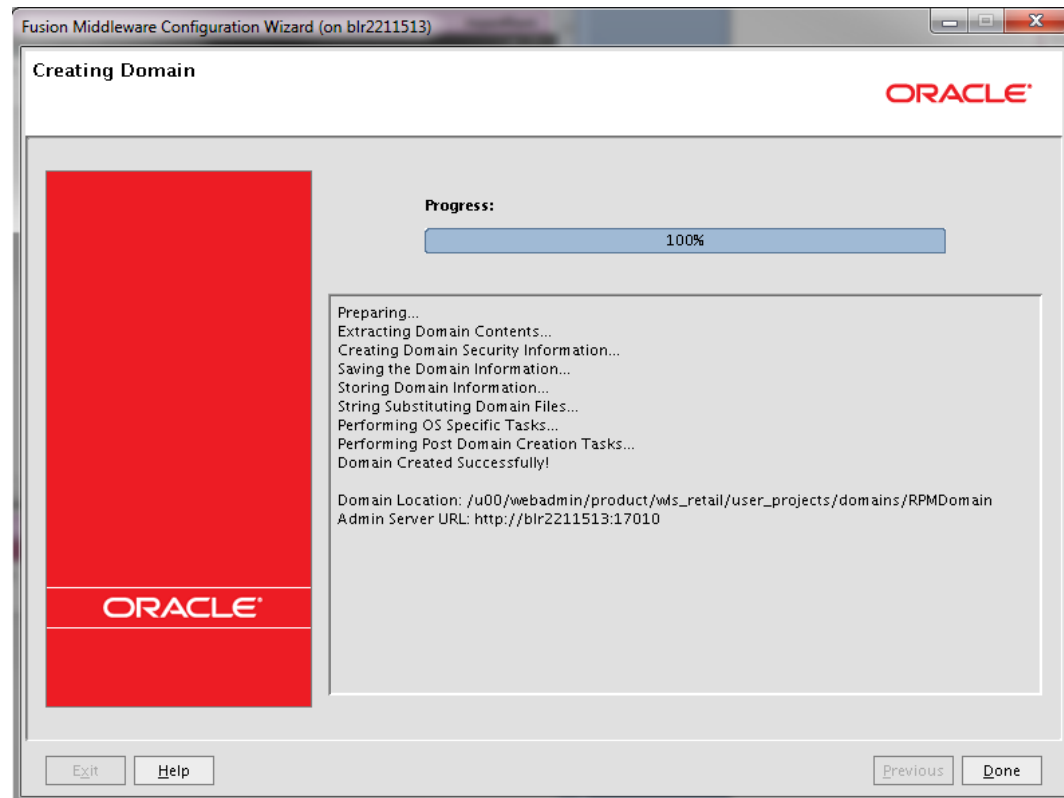
12. Select both the servers from the left and click on the arrow (towards right). The servers will move to the right and add to the Node manager. Click **Next**.



13. Click Create.



14. When the domain is successfully created, the following screen is displayed .Click **Done**.



Update the WebLogic.policy:

1. After the RPMdomain has been created, update `<WLS_HOME>/wlserver_10.3/server/lib/weblogic.policy` file with the information below.

Note: If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semi colon. Also, the AdminServer must be restarted for these changes to take effect.

Note: `<WEBLOGIC_DOMAIN_HOME>` in the example below is the full path of the WebLogic domain;
`<managed_server>` is the RPM managed server created.

```
grant codeBase
"file:<WEBLOGIC_DOMAIN_HOME>/servers/<managed_server>/tmp/_WL_user/<context_root>/" {
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore", "read,write,update,delete";
permission oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore.*", "read,write,update,delete";
};
```

```
grant codeBase
"file:<WEBLOGIC_DOMAIN_HOME>/servers/<managed_server>/cache/EJBCompilerCache/-
" {
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete";
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};
```

An example of the full entry that might be entered is:

```
grant codeBase
"file:/u00/webadmin/product/wls_retail/user_projects/domains/RPMdomain/servers
/rpm-server/tmp/_WL_user/rpml4/-" {
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore", "read,write,update,delete";
permission oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore.*", "read,write,update,delete";
};
```

```
grant codeBase
"file:/u00/webadmin/product/wls_retail/user_projects/domains/RPMdomain/servers
/rpm-server/cache/EJBCompilerCache/-" {
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete";
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};
```

Start the Node Manager

1. Start NodeManager from the server using the startNodeManager.sh at
\$WLS_HOME/wlserver_10.3/server/bin
2. Edit the nodemanager.properties file at the following location with the below values:
\$WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties
 - StartScriptEnabled=true
 - StartScriptName=startWebLogic.sh.
3. NodeManager must be restarted after making changes to the
nodemanager.properties file.

Note: The nodemanager.properties file is created after NodeManager is started for the first time. It will not be available before that point.

Start the AdminServer (admin console)

1. Start WebLogic Server from the <WEBLOGIC_DOMAIN_HOME>/bin

Example:

```
/u00/webadmin/product/wls_retail/user_projects/domains/RPMdomain/bin/startWebLogic.sh
```

2. Create boot.properties file under
<WEBLOGIC_DOMAIN_HOME>/servers/<AdminServer>/security

The file 'boot.properties' should have the following:

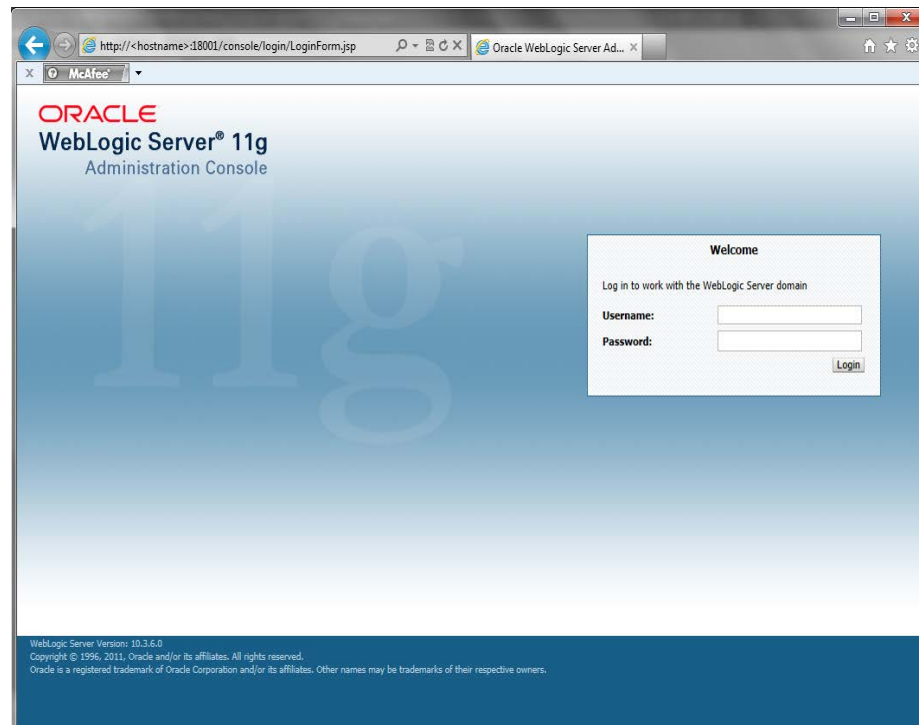
```
-----
username=weblogic
password=<password>
-----
```

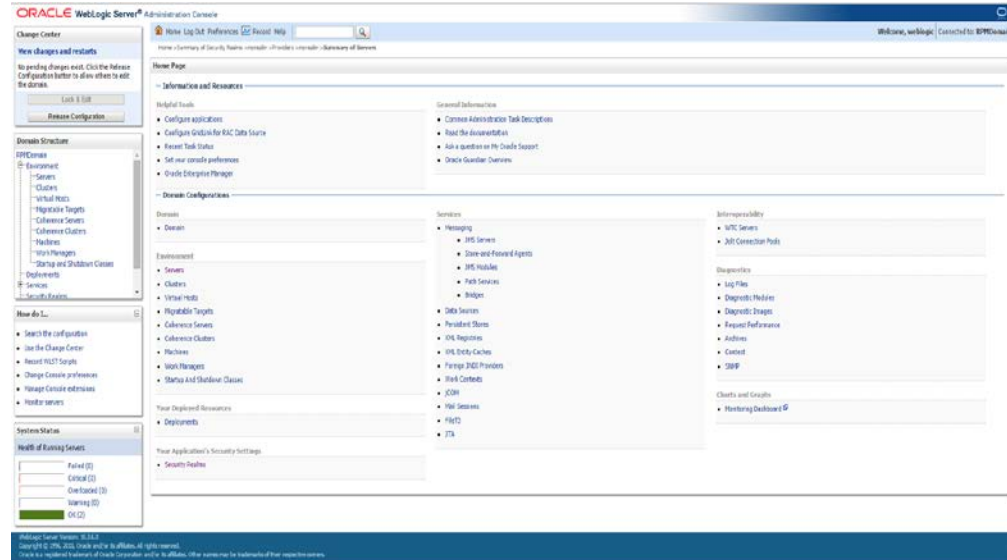
In the above, the password value is the password of WebLogic domain which is given at the time of domain creation.

3. Save the boot.properties file and restart the WebLogic server.
4. Login to the Admin console of the Domain

Example: <http://<hostname>:<port>/console>

In the below screen, provide username=weblogic and password=<weblogic password>

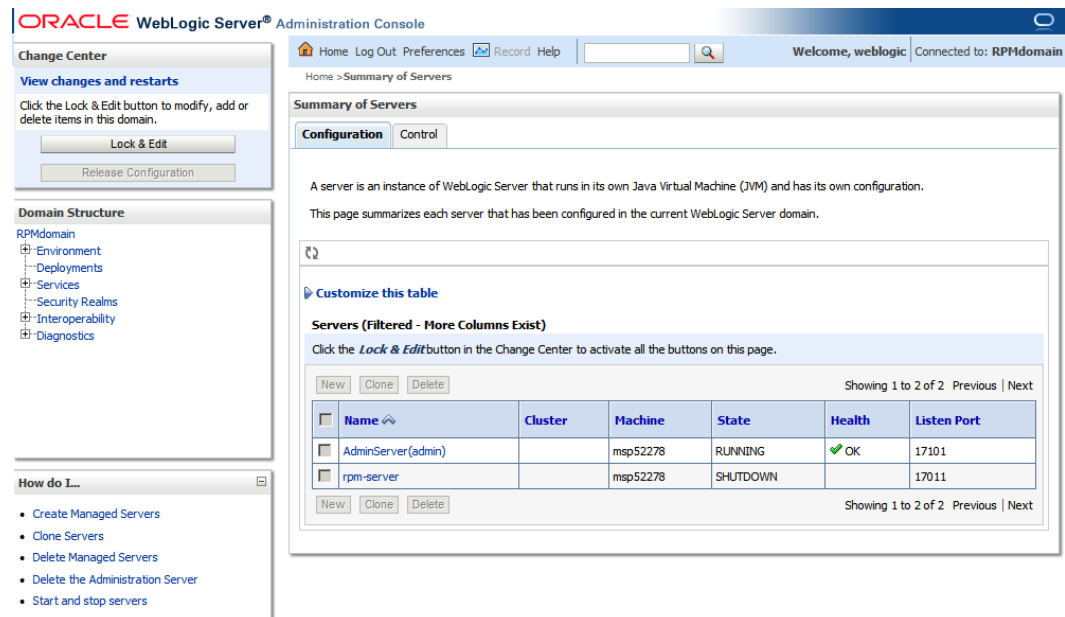




Start the Managed Server

After NodeManager is started, the managed servers can be started via the admin console.

1. Navigate to Environments -> Servers and click the Control tab. Select rpm-server and click **Start**.



Change the default (file based) Credential Store to use the Oracle Database

Create Required Schemas with RCU

The RPMdomain we just created by default uses a file based credential store for the wallet and policies. We need to change this to use the Oracle Database.

Some RCU database schemas are required to change the credential store of the RPMdomain from the default file based wallet to use an Oracle Database. Specifically, we will need to create the OPSS and MDS schemas.

The following steps will show you the creation of the database schemas required:

1. Download the RCU 11.1.1.7 zip file and extract it to a new folder named RCU 11.1.1.7. This folder (RCU 11.1.1.7) is used as RCU_HOME for the remainder of this guide. You may use a Windows version of RCU to create the schemas.
2. Go to <RCU_HOME>/bin and run ./rcu.
3. Select **Create** and click **Next**.



4. Select **Create Repository** and click **Next**.



5. Enter all the fields as explained below and click **Next**.
 - a. Host Name: Database server host name which Application will usePort: Database port (example: 1521)
 - b. Service Name Database name (example: dvols143)
 - c. Username: SYS
 - d. Password: <SYS password>

Repository Creation Utility - Step 2 of 7 : Database Connection Details

Database Connection Details

Database Type: Oracle Database

Host Name: hostname
For RAC database, specify VIP name or one of the Node name as Host name.
For SCAN enabled RAC database, specify SCAN host as Host name.

Port: 1521

Service Name: servicename

Username: SYS
User with DBA or SYSDBA privileges. Example:sys

Password:

Role: SYSDBA
One or more components may require SYSDBA role for the operation to succeed.

Messages:

Help < Back Next > Finish Cancel

6. Prerequisite requirements are verified and the following screen is displayed. Click **OK**.

Repository Creation Utility - Checking Prerequisites

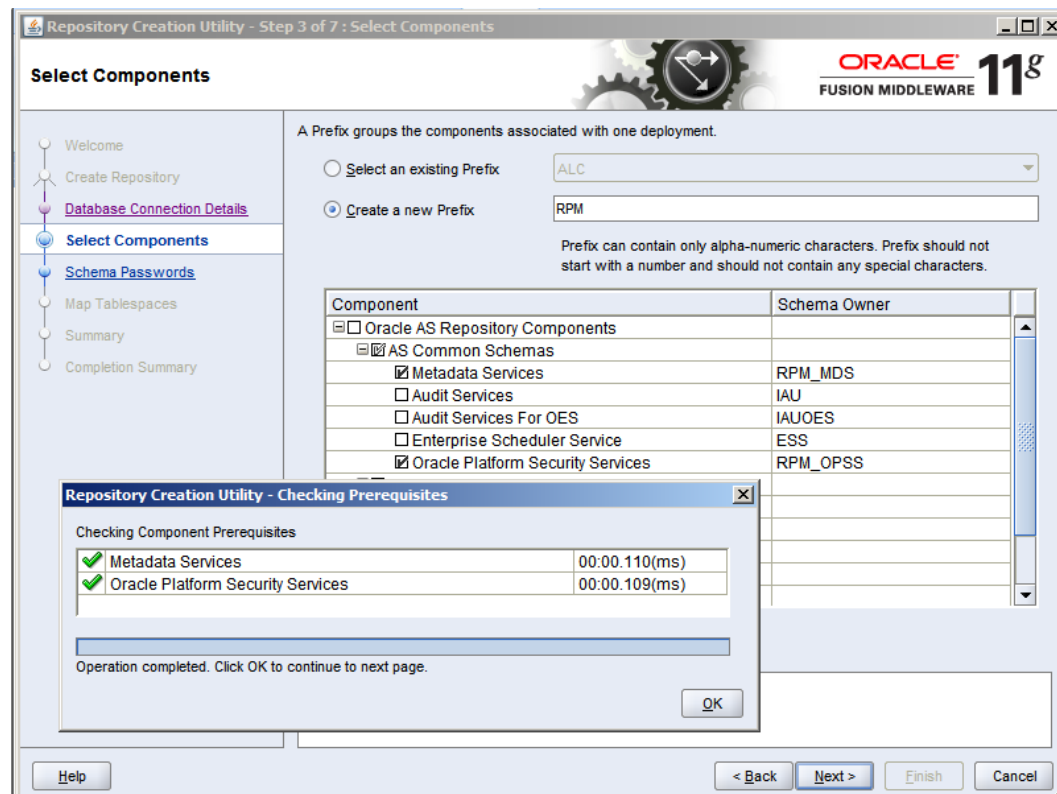
Checking Global Prerequisites

✓ Initializing repository configuration metadata	00:03.840(sec)
✓ Obtain properties of the specified database	00:00.100(ms)
✓ Check requirement for specified database	00:00.100(ms)
✓ Execute pre create operations	00:00.100(ms)

Operation completed. Click OK to continue to next page.

OK

7. Prerequisite requirements are verified and the following screen is displayed. Click **OK**. Expand Oracle AS Common Components and select the Metadata Services and Oracle Platform Security Services checkboxes as shown below. Enter a new prefix if needed (the example uses a prefix of “RPM”) and click **Next** then **OK** for the prerequisites check.



8. Enter and confirm your password and click **Next**.

Note: Make a note of the password you give here as it will be used later.

Repository Creation Utility - Step 4 of 7 : Schema Passwords

Schen **swords**

ORACLE 11g
FUSION MIDDLEWARE

Please enter the passwords for the main and additional (auxiliary) schema users. Password can contain alphabets, numbers and the following special characters: \$, #, _ . Password should not start with a number or a special character.

☒ Use same passwords for all schemas

Password:

Confirm Password:

☐ Use main schema passwords for auxiliary schemas

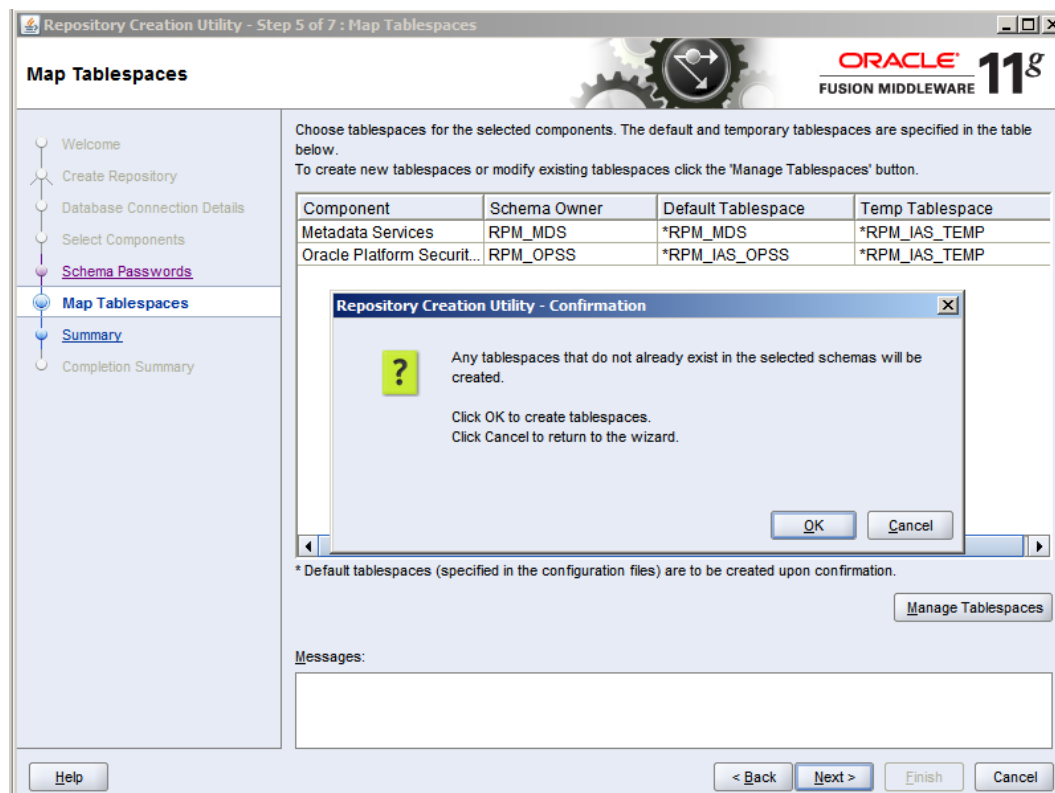
☐ Specify different passwords for all schemas

Component	Schema Owner	Schema Password	Confirm Password
Metadata Services	RPM_MDS		
Oracle Platform Security Services	RPM_OPSS		

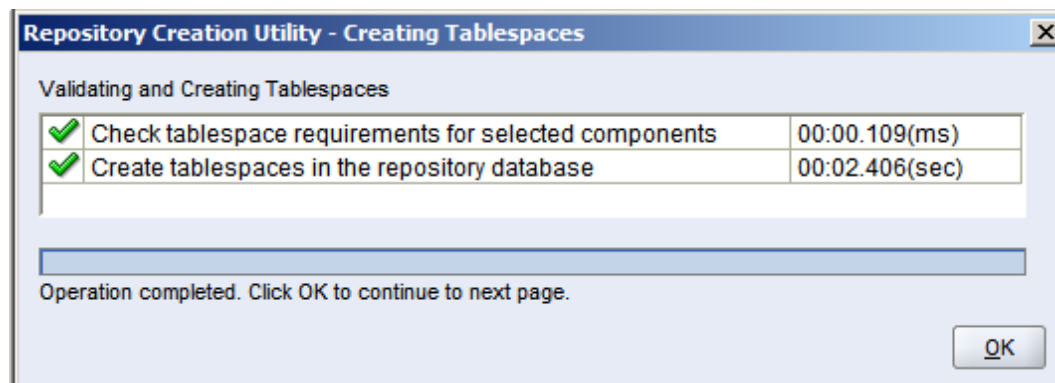
Messages:

Help < Back Next > Finish Cancel

9. Click **Next**, then **OK** when it states it is going to create the tablespaces if they are needed.



10. Click **OK** when tablespace creating and validation has completed.



11. Click **Create** to make the schemas.

Repository Creation Utility - Step 6 of 7 : Summary

Summary

Database details:

Host Name: msp52410.us.oracle.com
Port: 1521
Service Name: DVOLS143
Connected As: SYS
Operation: Create

Prefix for (prefixable) Schema Owners: RPM

Component	Schema Owner	Tablespace Type	Tablespace Name
Metadata Services	RPM_MDS	Default	RPM_MDS
		Temp	RPM_IAS_TEMP
		Additional	None
Oracle Platform Security Services	RPM_OPSS	Default	RPM_IAS_OPSS
		Temp	RPM_IAS_TEMP
		Additional	None

Help < Back Next > Create Cancel

12. Schemas are created, click **Close** to exit RCU.



Set up OPSS Schema Data source in WebLogic domain

Follow the below steps to set up the data source with OPSS schema in WebLogic domain (RPMdomain).

1. Login to the Admin console and go to Services -> Data Sources.

The screenshot shows the Oracle WebLogic Server Administration Console. On the left, the 'Change Center' panel is visible, showing 'View changes and restarts' and 'Lock & Edit' buttons. Below it is the 'Domain Structure' tree, which is expanded to 'Services' > 'Data Sources'. The main panel displays the 'Summary of JDBC Data Sources' page. It includes a 'Configuration' tab and a 'Monitoring' tab. A description of a JDBC data source is provided. Below the description, there is a 'Customize this table' link and a table titled 'Data Sources (Filtered - More Columns Exist)'. The table has columns for 'Name', 'Type', 'JNDI Name', and 'Targets'. The table is currently empty, showing 'There are no items to display'. Navigation links like 'New', 'Delete', 'Previous', and 'Next' are present.

2. Click **Lock & Edit** then click **New** -> **Generic Data Source**.

The screenshot shows the 'Create a New JDBC Data Source' wizard in the Oracle WebLogic Server Administration Console. The 'Change Center' panel on the left shows 'No pending changes exist. Click the Release Configuration button to allow others to edit the domain.' The 'Domain Structure' tree is expanded to 'Services' > 'Data Sources'. The main panel is titled 'Create a New JDBC Data Source' and includes 'Back', 'Next', 'Finish', and 'Cancel' buttons. The 'JDBC Data Source Properties' section states: 'The following properties will be used to identify your new JDBC data source. * Indicates required fields'. The first step asks 'What would you like to name your new JDBC data source?' with a text field containing 'JDBC Data Source-0'. The second step asks 'What JNDI name would you like to assign to your new JDBC Data Source?' with an empty text field. The third step asks 'What database type would you like to select?' with a dropdown menu showing 'Oracle'. Navigation buttons 'Back', 'Next', 'Finish', and 'Cancel' are at the bottom.

3. Enter the details and click **Next**.

- Name: OPSSDataSource
- JNDI Name: jndi/OPSSDS
- Database Type: Oracle

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Home > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

* Indicates required fields

What would you like to name your new JDBC data source?

Name: OPSSDataSource

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name: jndi/OPSSDS

What database type would you like to select?

Database Type: Oracle

Back Next Finish Cancel

4. Select Oracle's Driver (Thin XA) for Instance connections; Versions: 9.0.1 and later and click **Next**.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: APPDomain

Home > Summary of Security Realms > myrealm > Users and Groups > myrealm > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

Database Type: Oracle

What database driver would you like to use to create database connections? Note: * indicates that the driver is explicitly supported by Oracle WebLogic Server.

Database Driver: *Oracle's Driver (Thin XA) for Instance connections, Versions 9.0.1 and later

Back Next Finish Cancel

5. Click **Next**.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: APPDomain

Home > Summary of Security Realms > myrealm > Users and Groups > myrealm > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

Transaction Options

You have selected an XA JDBC driver to use to create database connection in your new data source. The data source will support global transactions and use the 'Two-Phase Commit' global transaction protocol. No other transaction configuration options are available.

Back Next Finish Cancel

6. Enter the details of the OPSS schema we just created and click **Next**.
 - Database Name: <DBName >(i.e: dvols143)
 - Url – jdbc:oracle:thin:@<hostname>:<1521>/<servicename>
 - Host Name: <DB Host Name>
 - Port: 1521
 - Database User Name: RPM_OPSS (This is the OPSS schema which has been created using RCU earlier in this document.)
 - Password: <password> (Password given at the time of OPSS schema creation)

The screenshot shows the Oracle WebLogic Server Administration Console. The main window displays the 'Create a New JDBC Data Source' wizard. The 'Test Database Connection' step is active, showing the following fields:

- Driver Class Name:** oracle.jdbc.xa.client.OracleDriver
- URL:** jdbc:oracle:thin:@msp024
- Database User Name:** RPM_OPSS
- Password:** (masked with asterisks)
- Confirm Password:** (masked with asterisks)
- Properties:** uac=OPSS

The left sidebar shows the 'Domain Structure' tree with the 'Services' node selected. The 'System Status' section shows the 'Health of Running Servers' with a green bar indicating 'OK'.

7. Click **Test Configuration** and make sure you can connect to the schema successfully. Click **Next** if you can connect. Click **Back** if it does not connect and check your settings like URL syntax (jdbc:oracle:thin:@<hostname>:<1521>/<servicename>) , username and password.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Change Center
View changes and restarts
No pending changes exist. Click the Release Configuration button to allow others to edit the domain.
Look & Edit
Release Configuration

Domain Structure
RPMdomain
Environment
Deployments
Services
Messaging
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
XML Registries
XML Entity Caches
JCOM
Mail Sessions
File T3

How do I...
Create JDBC generic data sources
Create LIR-enabled JDBC data sources

System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)

Create a New JDBC Data Source
Test Configuration Back Next Finish Cancel

Test Database Connection
Test the database availability and the connection properties you provided.
What is the full package name of JDBC driver class used to create database connections in the connection pool?
(Note that this driver class must be in the classpath of any server to which it is deployed.)
Driver Class Name: oracle.jdbc.OracleDriver
What is the URL of the database to connect to? The format of the URL varies by JDBC driver.
URL: jdbc:oracle:thin:@msp524
What database account user name do you want to use to create database connections?
Database User Name: RPM_OPSS
What is the database account password to use to create database connections?
(Notes: for secure password management, enter the password in the Password field instead of the Properties field below)
Password:
Confirm Password:
What are the properties to pass to the JDBC driver when creating database connections?

8. Target all the servers (AdminServer & rpm-server) and click **Finish**.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Change Center
View changes and restarts
No pending changes exist. Click the Release Configuration button to allow others to edit the domain.
Look & Edit
Release Configuration

Domain Structure
RPMdomain
Environment
Deployments
Services
Messaging
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
XML Registries
XML Entity Caches
JCOM
Mail Sessions
File T3

How do I...
Create JDBC generic data sources
Create LIR-enabled JDBC data sources

System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)

Create a New JDBC Data Source
Back Next Finish Cancel

Select Targets
You can select one or more targets to deploy your new JDBC data source. If you don't select a target, the data source will be created but not deployed. You will need to deploy the data source at a later time.
Servers
AdminServer
rpm-server
Back Next Finish Cancel

9. Click **Activate Changes** to get them incorporated into the domain.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Change Center
View changes and restarts
Pending changes exist. They must be activated to take effect.
Activate Changes
Undo All Changes

Domain Structure
RPMdomain
Environment
Deployments
Services
Messaging
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
XML Registries
XML Entity Caches
JCOM
Mail Sessions
File T3

How do I...
Create JDBC generic data sources
Create LIR-enabled JDBC data sources

System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)

Summary of JDBC Data Sources
Configuration Monitoring
A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. Applications can look up a data source on the JNDI tree and then borrow a database connection from a data source.
This page summarizes the JDBC data source objects that have been created in this domain.
Customize this table
Data Sources (Filtered - More Columns Exist)
New Delete
Showing 1 to 1 of 1 Previous Next
Name Type JNDI Name Targets
OPSSDataSource Generic jndi/OPSSDS AdminServer, rpm-server
New Delete
Showing 1 to 1 of 1 Previous Next

10. Once the changes have been incorporated into the domain, a message is displayed notifying you that the changes have been activated.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences [Add](#) Record Help

Welcome, weblogic Connected to: RPMdomain

Home > Summary of JDBC Data Sources

Messages

✓ All changes have been activated. No restarts are necessary.

Summary of JDBC Data Sources

Configuration Monitoring

A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. Applications can look up a data source on the JNDI tree and then borrow a database connection from a data source.

This page summarizes the JDBC data source objects that have been created in this domain.

Customize this table

Data Sources (Filtered - More Columns Exist)

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

Name	Type	JNDI Name	Targets
OPSSDataSource	Generic	jndi/OPSSDS	AdminServer, rpm-server

Showing 1 to 1 of 1 Previous Next

How do I...
• Create JDBC generic data sources
• Create JDBC GridLink data sources
• Create JDBC multi data sources

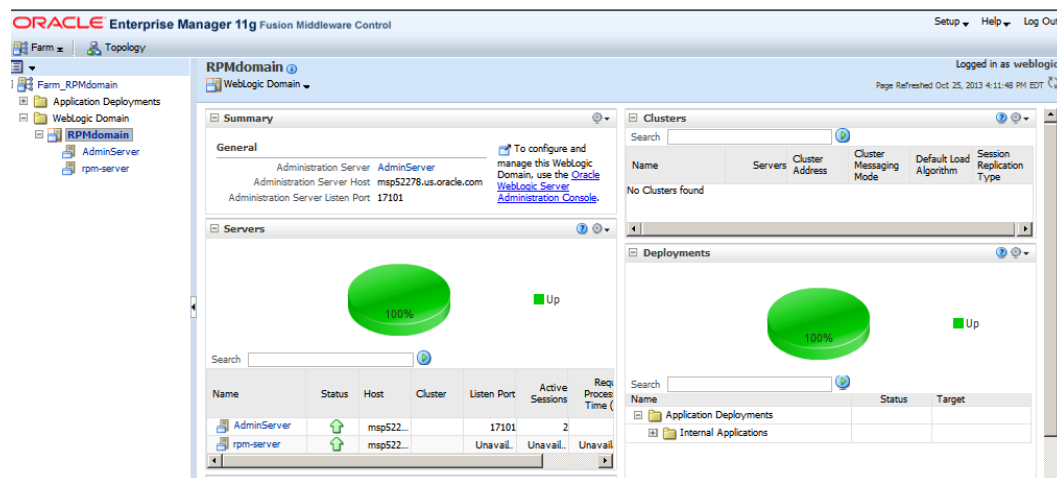
Associate Policy Store to Database

Follow the steps below to re-associate the domain policy store from file based to using the database:

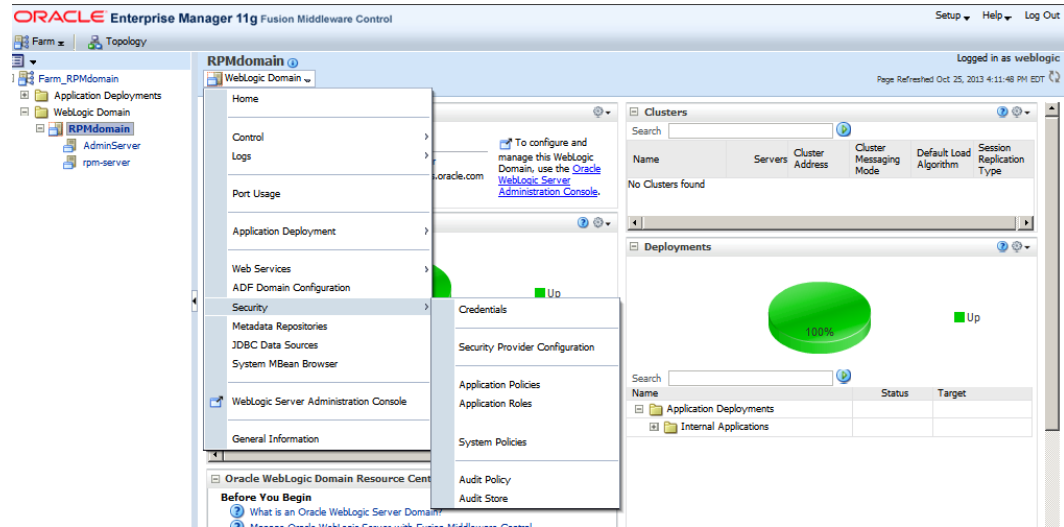
1. Login to the WebLogic EM console (i.e.: <http://<Host Name >:17101/em>).



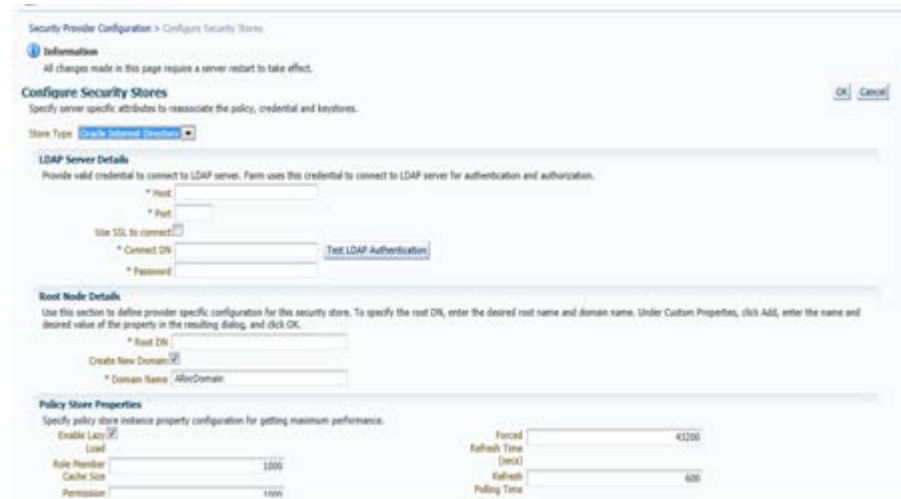
2. Expand the WebLogic Domain and click the **RPMdomain**.



3. Select the dropdown WebLogic Domain->Security->Security Provider Configuration.



4. Click Change Store Type.



5. Select Oracle Database in the Store Type drop down.

Security Provider Configuration > Configure Security Stores

Information
All changes made in this page require a server restart to take effect.

Configure Security Stores
Specify server specific attributes to reassociate the policy, credential and keystores.

Store Type: **Oracle Internet Directory**

LDAP Server: **Oracle Internet Directory**

Provide valid credential to connect to LDAP server. Fann uses this credential to connect to LDAP server for authentication and authorization.

* Host:

* Port:

Use SSL to connect: ☐

* Connect DN: **Test LDAP Authentication**

* Password:

Root Node Details
Use this section to define provider specific configuration for this security store. To specify the root DN, enter the desired root name and domain name. Under Custom Properties, click Add, enter the name and desired value of the property in the resulting dialog, and click OK.

* Root DN:

Create New Domain: ☒

* Domain Name:

Policy Store Properties
Specify policy store instance property configuration for getting maximum performance.

Enable Ldap: ☒

Cache Size:

Permissions:

Period:

Refresh Time (secs):

Polling Time:

Update Cache: ☒

Incrementally for Management: ☐

6. Click **Select** and select jndi/OPSSDS JNDI name. Click **OK**.

Security Provider Configuration > Configure Security Stores

Specify server specific attributes to reassociate the policy, credential and keystores.

Store Type: **Oracle Database**

Database Server Details

* Data Source JNDI Name:

Data Source Access

Protected Data Source: ☐

User Name:

Password:

Root Node Details
Use this section to define provider specific configuration for this security store. To specify the root DN, enter the desired root name and domain name. Under Custom Properties, click Add, enter the name and desired value of the property in the resulting dialog, and click OK.

* Root DN:

Create New Domain: ☒

* Domain Name:

Policy Store Properties
Specify policy store instance property configuration for getting maximum performance.

Enable Ldap: ☒

Cache Size:

Permissions:

Period:

Refresh Time (secs):

Polling Time:

Update Cache: ☒

Incrementally for Management: ☐

Select JNDI Databases

JNDI Name	Data Source JNDI Name
jndi/OPSSDS	OPSS Database
jndi/Custom...	mtb-CustomPortaIDS

OK Cancel

7. Enter the values:

- Provide OPSS schema username and password configured while creating the schema with rcu
- Root DN= cn=RMPolicies
- Select 'Create New Domain'
- Domain Name=RPMdomain (This must be the domain name which has been created earlier in this document)

8. Click OK.

RPMDomain @ WebLogic Domain

Logged in as **weblog**
Page Refreshed Nov 12, 2014 11:42:45 PM EDT

Security Provider Configuration > Configure Security Stores

Information
All changes made in this page require a server restart to take effect.

Configure Security Stores
Specify server specific attributes to reassociate the policy, credential and keystores.

Store Type: **Oracle Database**

Database Server Details
* Data Source JNDI Name: **jms/CPSSDS**

Data Source Properties
Driver Class Name: **oracle.jdbc.OracleDriver**
Database URL: **jdbc:oracle:thin:@map12013.us.oracle.com:1521/DOLSP01APP**
* User Name: **RPM_CPSS**
* Password: *********
* Confirm Password: *********
ODBC Data Source Name:

Data Source Access
Protected Data Source: ☐
User Name:
Password:

Root Node Details
Use this section to define provider specific configuration for this security store. To specify the root DN, enter the desired root name and domain name. Under Custom Properties, click Add, enter the name and desired value of the property in the resulting dialog, and click OK.
* Root DN:
Create New Domain: ☒
* Domain Name: **RPMDomain**

Policy Store Properties
Specify policy store instance property configuration for getting maximum performance.
Enable Lazy Load: ☐

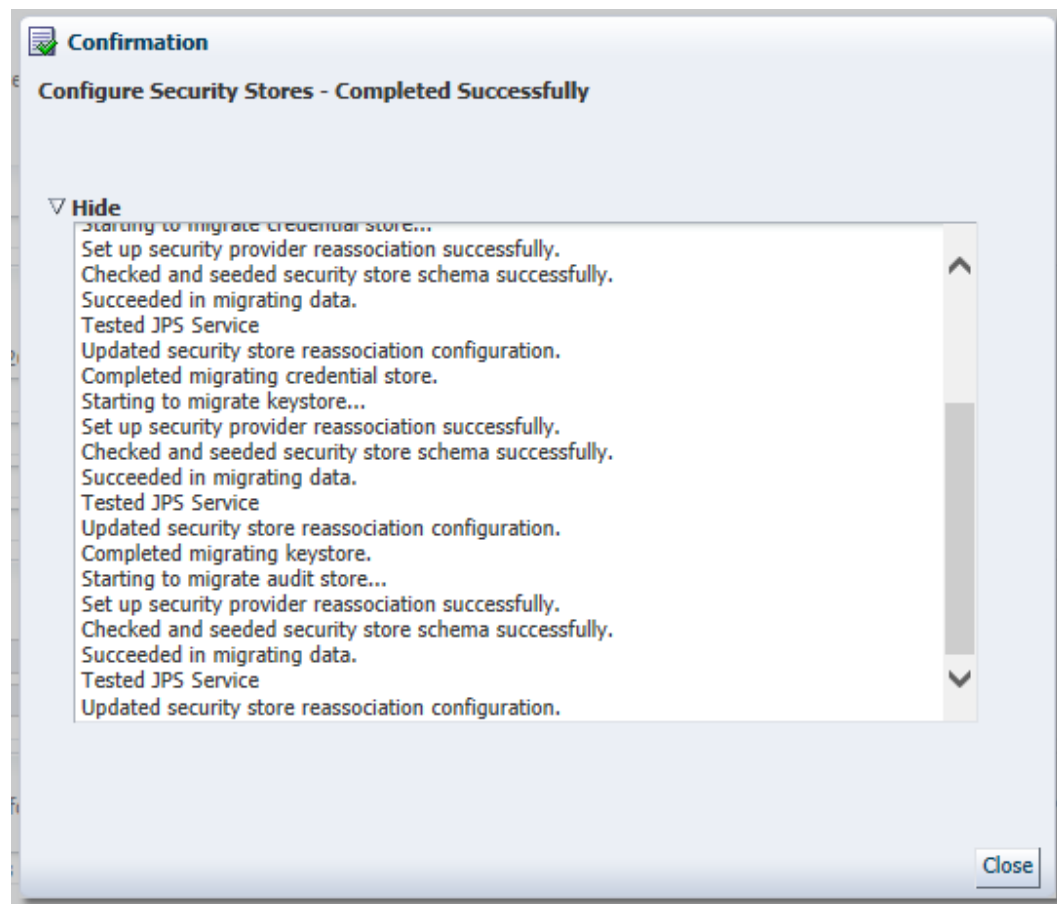
Forced Refresh: 41000

9. Click Yes.

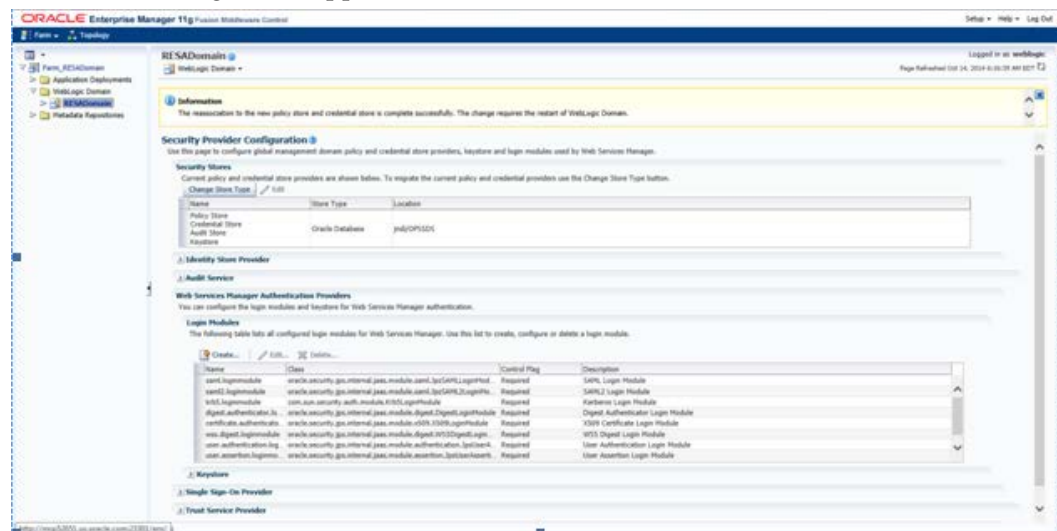
Confirmation

The change requires the restart of management domain. Are you sure you want to change security store provider?

10. The message Configure Security Stores – Completed Successfully appears. Click Close.



The following screen appears:

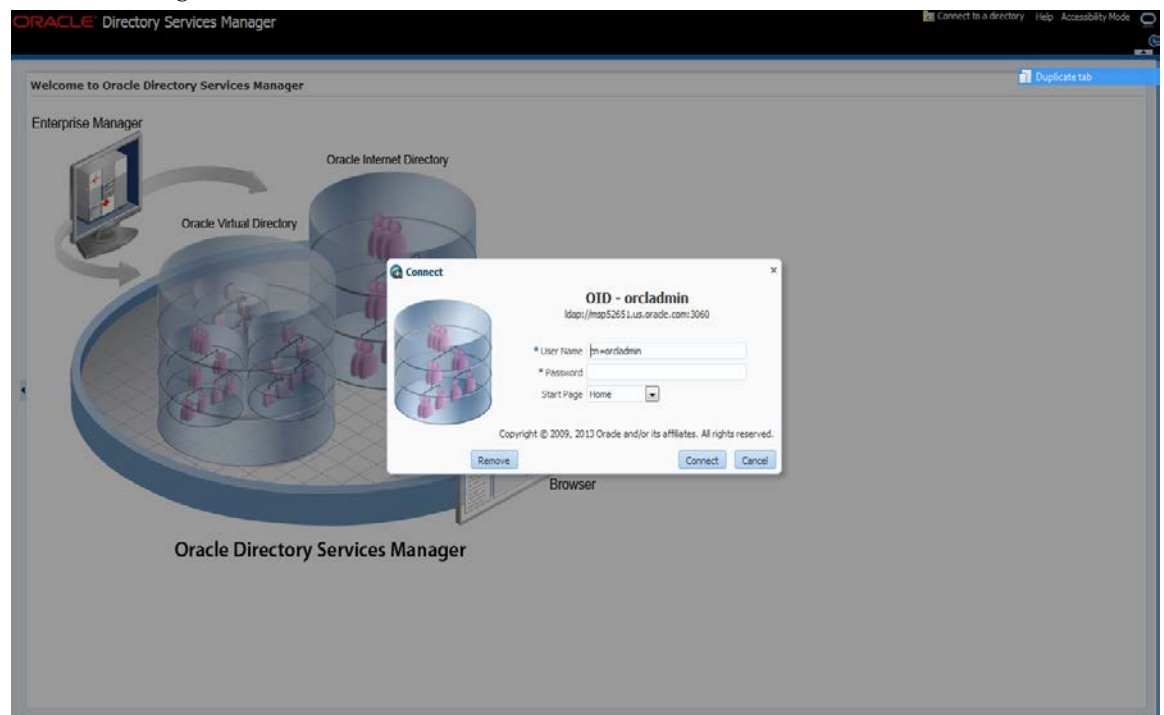


11. Restart the WebLogic domain.
12. Restart the RPMdomain to get the change to take effect.

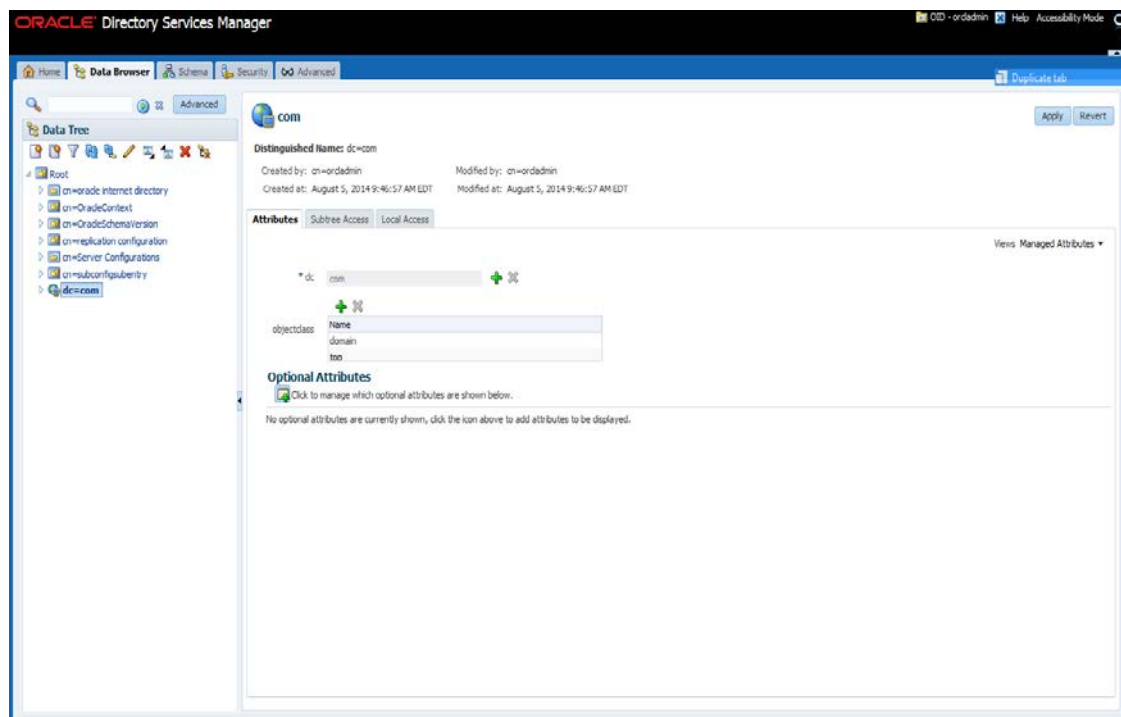
Configure LDAP authentication Preinstallation Steps (Initial Login to RPM)

In order to Login to RPM after the installation is done, you need to complete the following pre-installation steps.

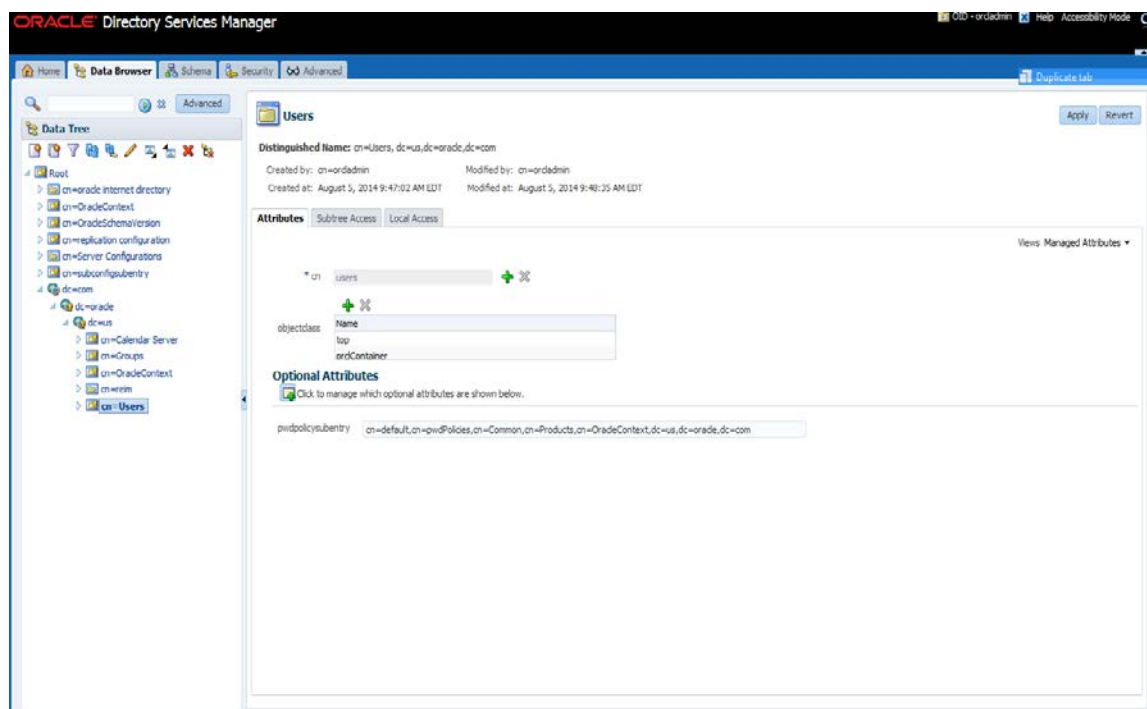
1. Make sure that you have access to a working LDAP server.
2. Create an LDAP connection user with the necessary rights to do sub-tree searches on your users and groups respectively. This user can be named anything but "RPM.ADMIN" is used in this document. This same user should be given as an input for 'Search User DN' on the 'LDAP Directory Server Details' screen while installing the RPM application. This is the user which RPM uses to login to LDAP and perform the necessary search in the LDAP.
3. Follow the below steps to create the 'example:RPM.ADMIN' user.
 - a. Open your OID connection by launching ODSM (Oracle Directory Services Manager).



- b. From the OID Connect dialog, click the Connect button.
- c. From the Oracle Internet Directory Welcome Screen, select the Data Browser tab. The Data Browser tree shows how to find the "cn=Users" element.



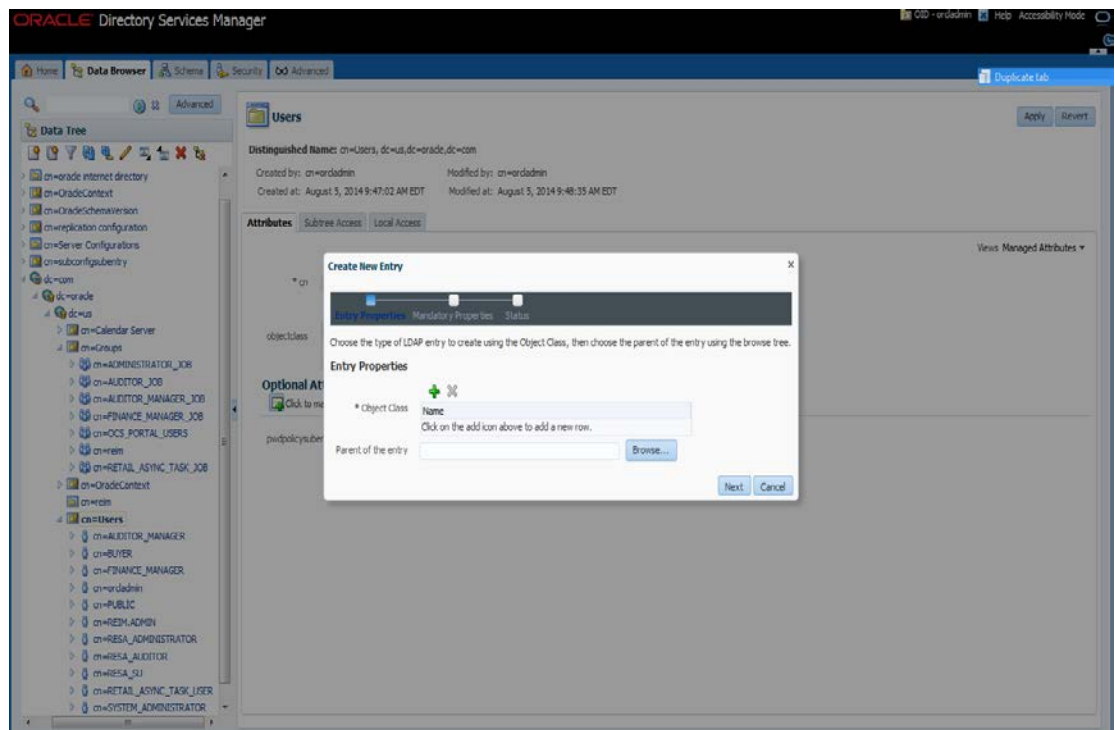
- d. From the Data Tree panel of the ODSM screen, navigate to “Users” branch.



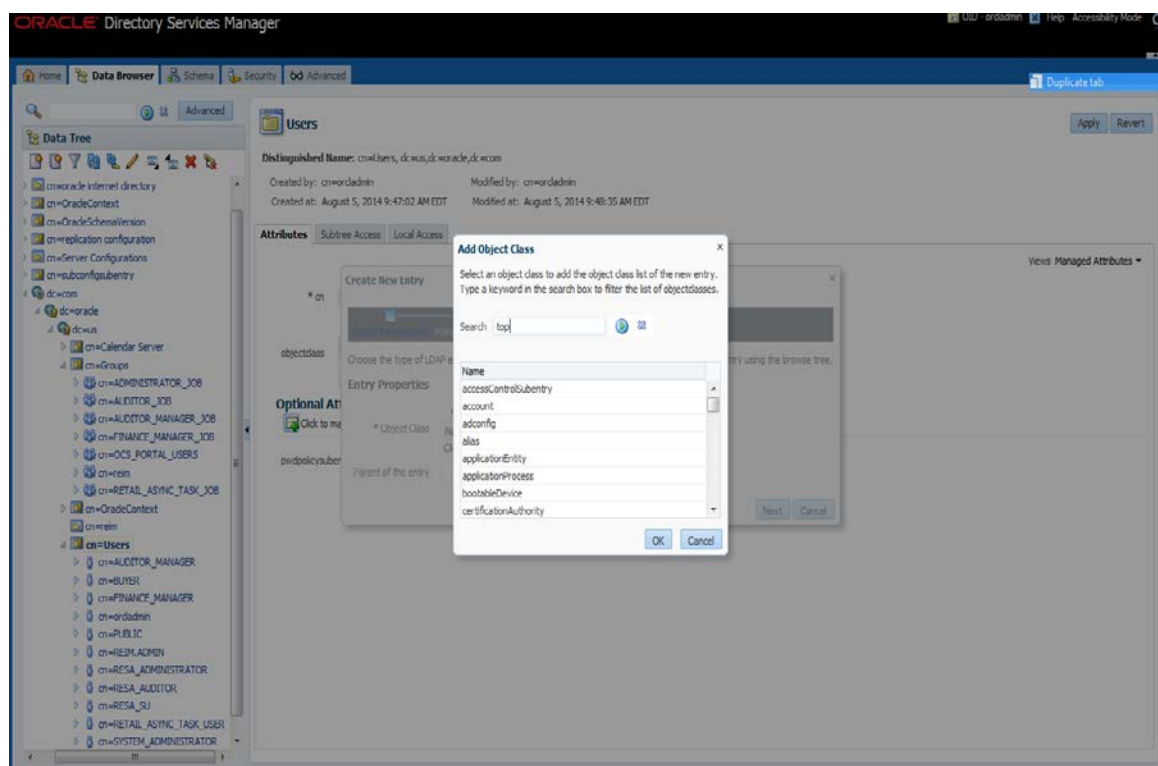
- e. On the “Users” screen , press right mouse button with “cn=Users” highlighted and select “Create” from the drop down menu panel

- f. In the Object Class field, click the + icon to add the following Object Classes
- top
 - orclContainer
 - organizationalperson
 - orcluser
 - person
 - orcluserv2
 - Inetorgperson

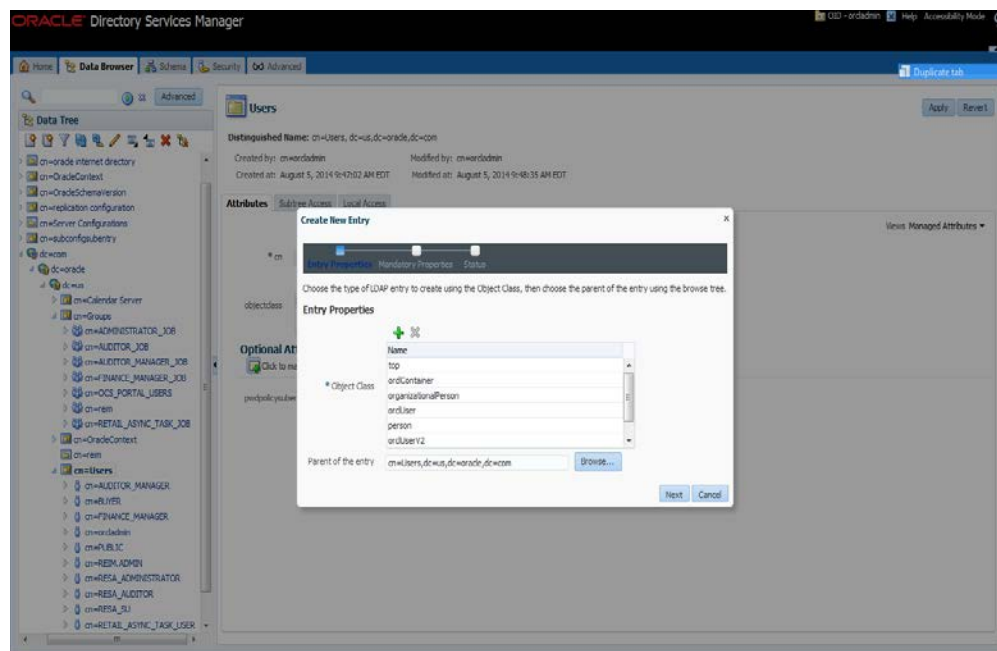
Note: Only one Object Class can be added at a time so the next few steps will need to be repeated until all of the Object Classes have been added.



- g. From the Add Object Class menu, select the “top” object class.

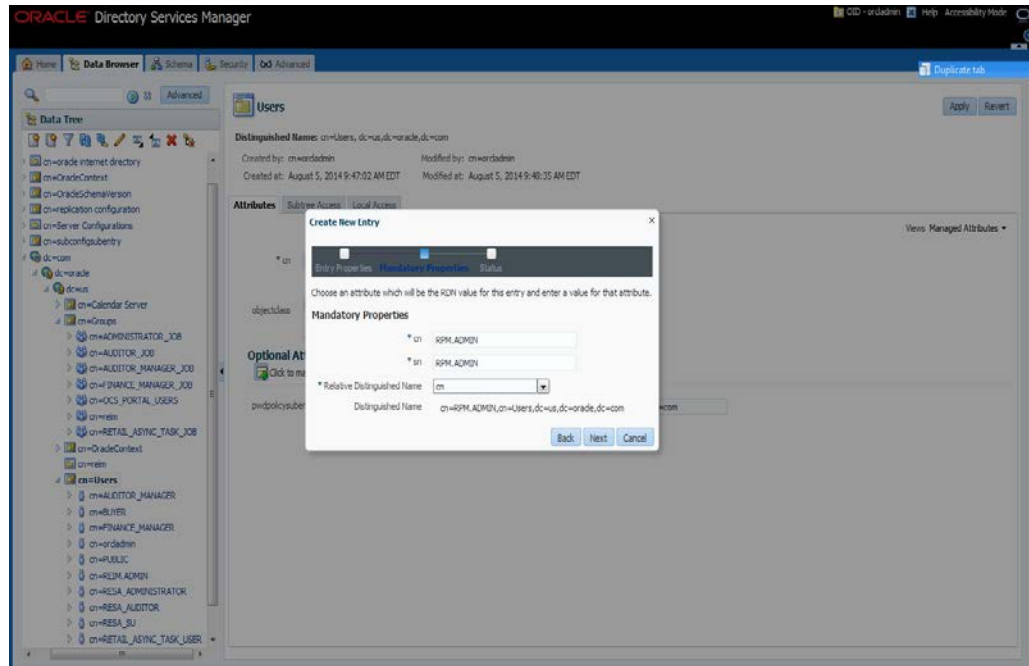


- h. From the Add Object Class menu, select the “orclContainer” object class.
- i. When finished adding in all the Object Classes the screen will look as it does below. Then In the Parent of the Entry field enter the following:
cn=Users,dc=us,dc=oracle,dc=com

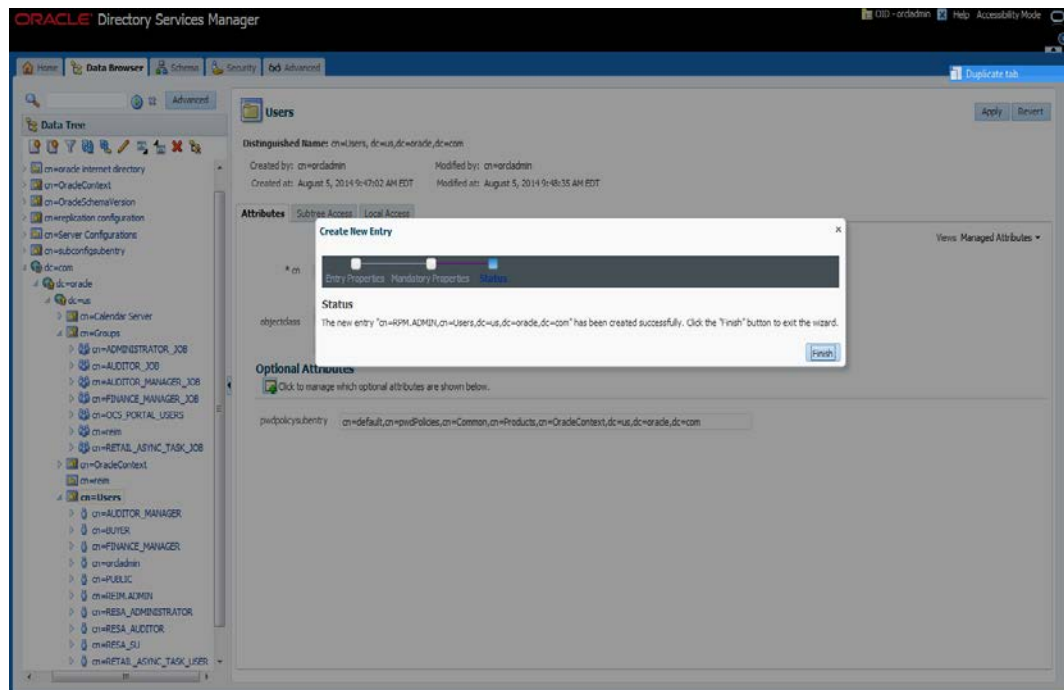


- j. Click Next. The Mandatory Properties dialog is displayed.

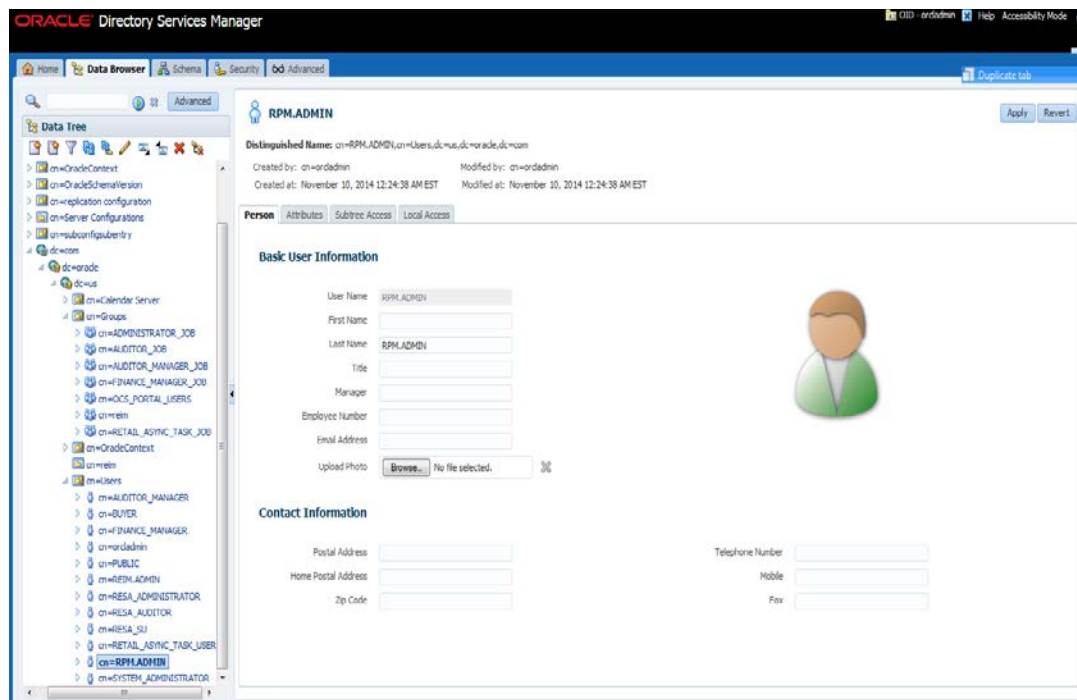
- k. From Mandatory Properties dialog, enter in the following and click next:
- cn= RPM.ADMIN
 - sn= RPM.ADMIN
 - Relative Distinguished = cn



- l. Make sure the information on screen is correct. Press **Finish** to create the “RPM.ADMIN” user.



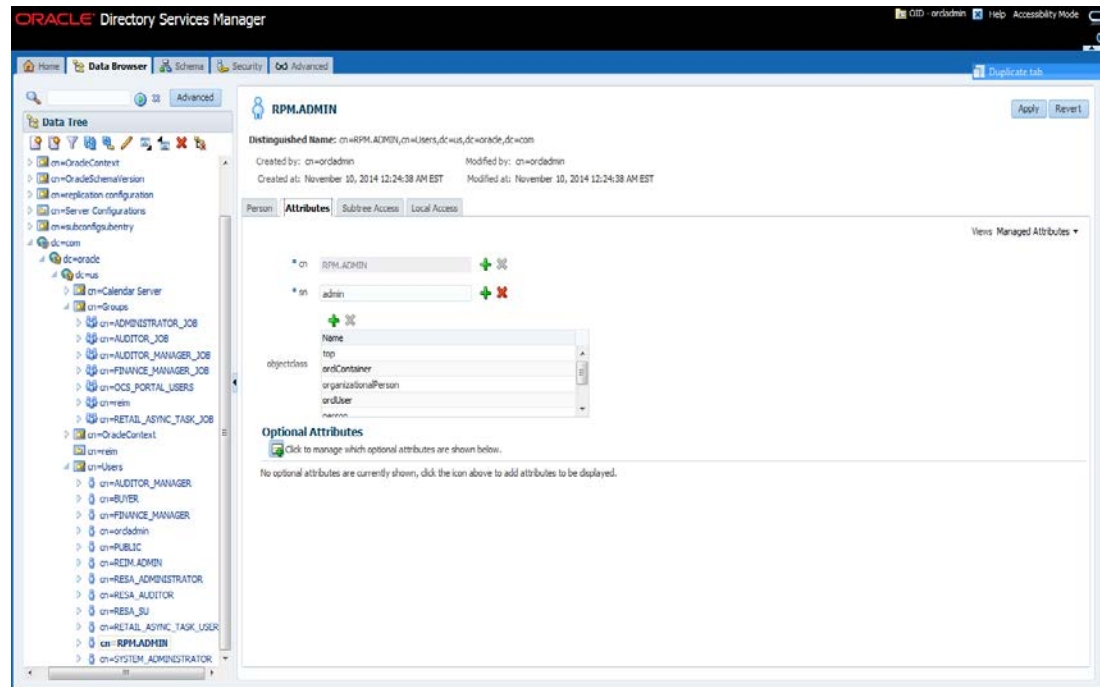
When the “RPM.ADMIN” user is created a screen similar to the one below is displayed when clicking on the new RPM.ADMIN user.



m. On the Person tab and enter the following Basic User Information:

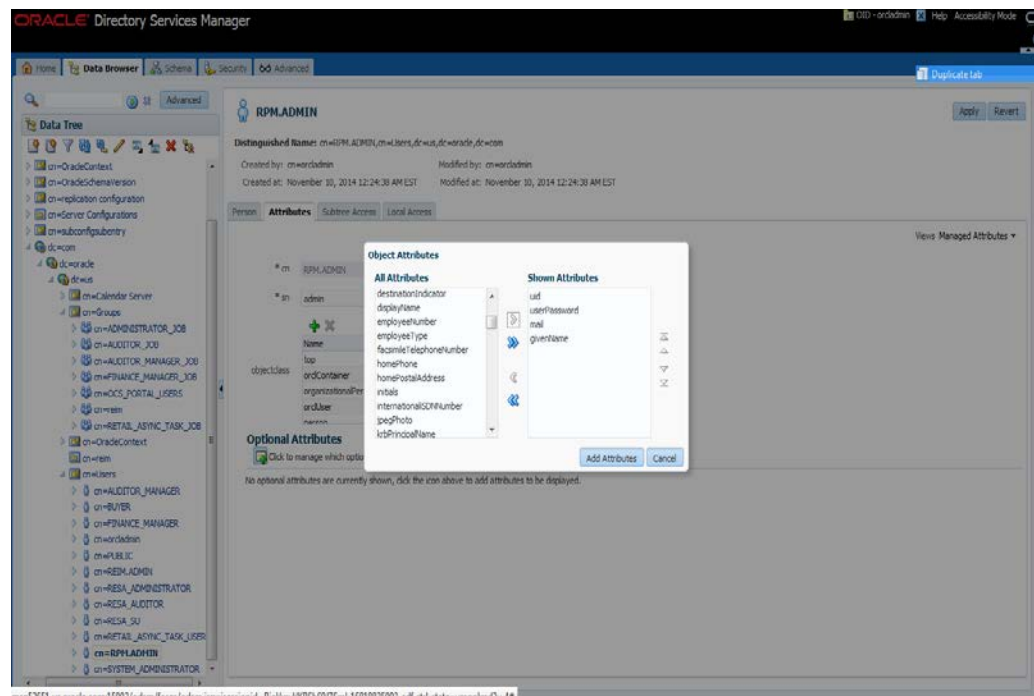
- First Name: <rpm>
- Last Name: <admin>
- Email Address: <rpm.admin@mycompany.com>

n. Click the Attributes tab.



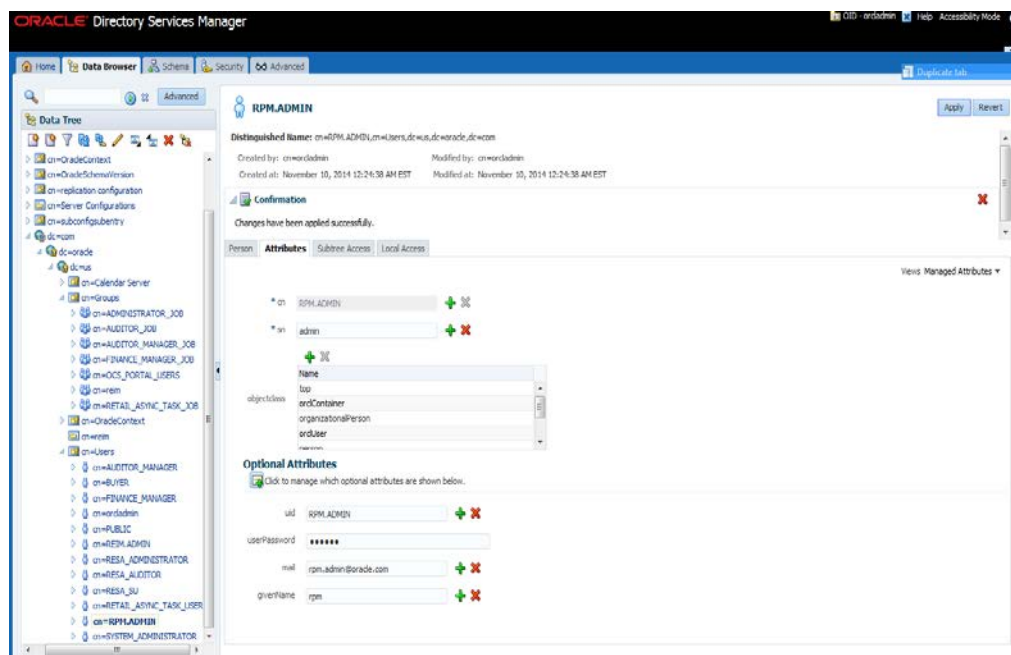
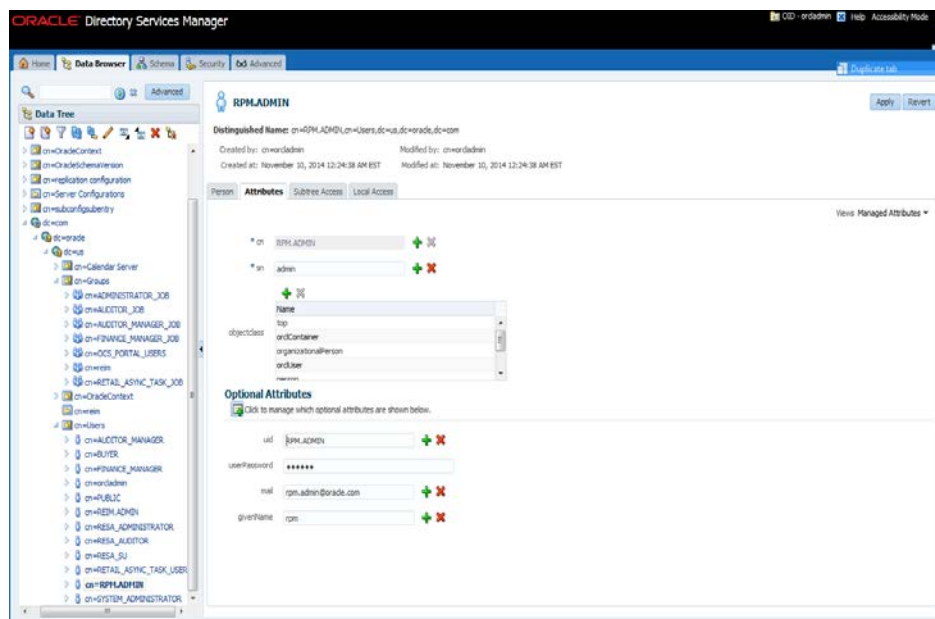
o. Click on the Add Optional Attributes button and select:

- Given Name: <rpm>
- Mail: <rpm.admin@mycompany.com>
- Uid: RPM.ADMIN
- User Password: <password>



p. Enter the following information and click Apply:

- Given Name: <rpm>
- Mail: <rpm.admin@mycompany.com>
- Uid: RPM.ADMIN
- User Password: <password>

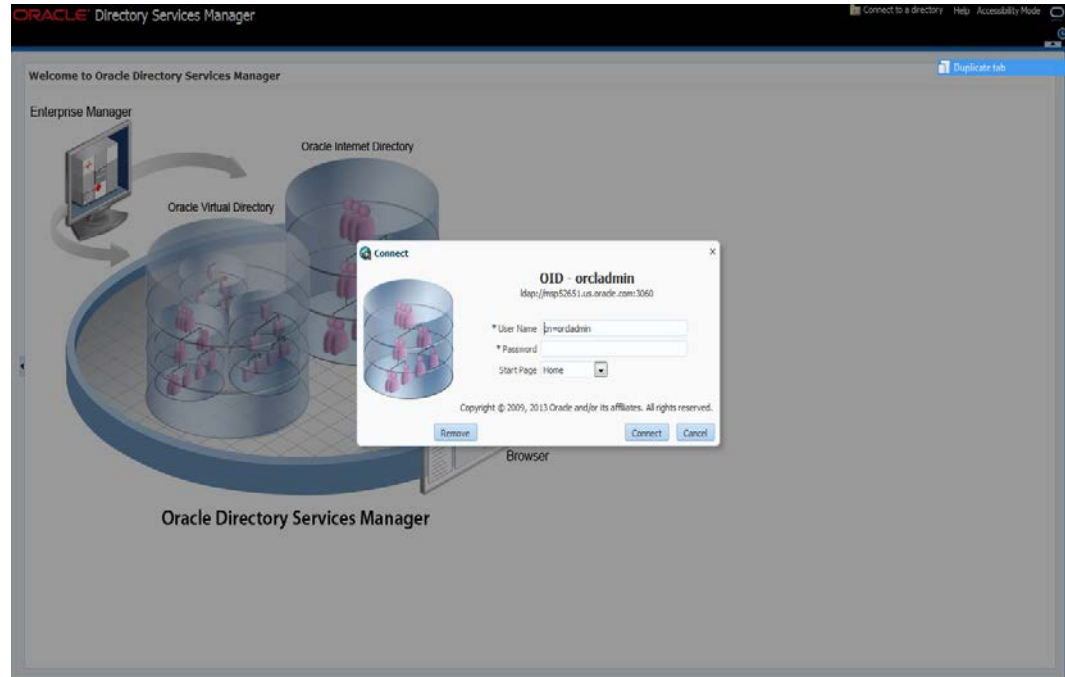


4. Create the Application Admin user who will have access (Login) to RPM.
 - a. If you are installing other Merchandising applications you should have already created RETAIL.USER. If you do not have RETAIL.USER already created in LDAP, create "RETAIL.USER" following the same procedure described for creating the RPM.ADMIN user above.<<already present

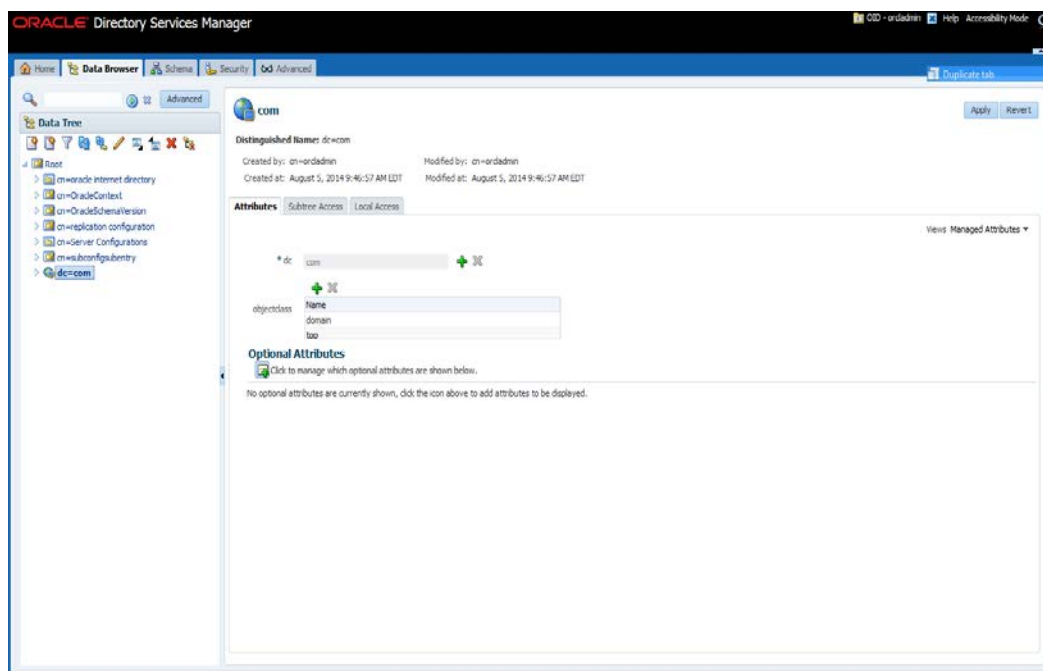
5. Create a Group called "rpm_secure_users". All users need to be a member of this group in order to login to the RPM application.

Note: The RPM code looks for a group named "rpm_secure_users" so it is imperative that the group be named "rpm_secure_users".

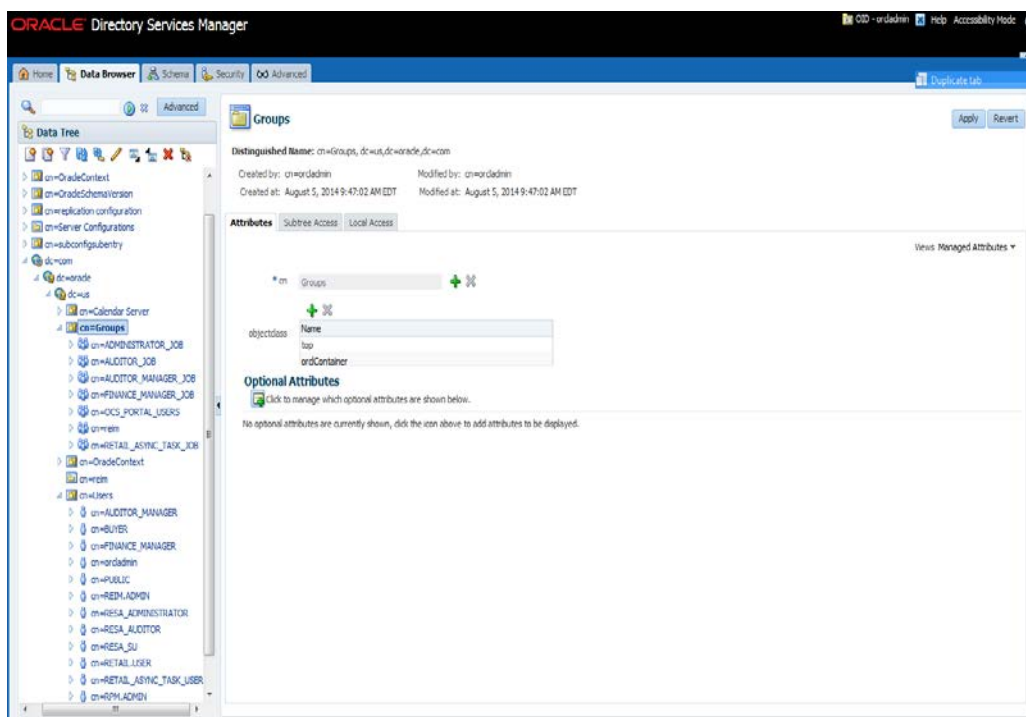
6. Example: Using OID 11.1.1.7, the steps to follow are:
 - a. Open your OID connection by launching odsm (Oracle Directory Services Manager). A screen similar to the following is displayed.
 - b. Click Connect to a directory and select your OID directory.



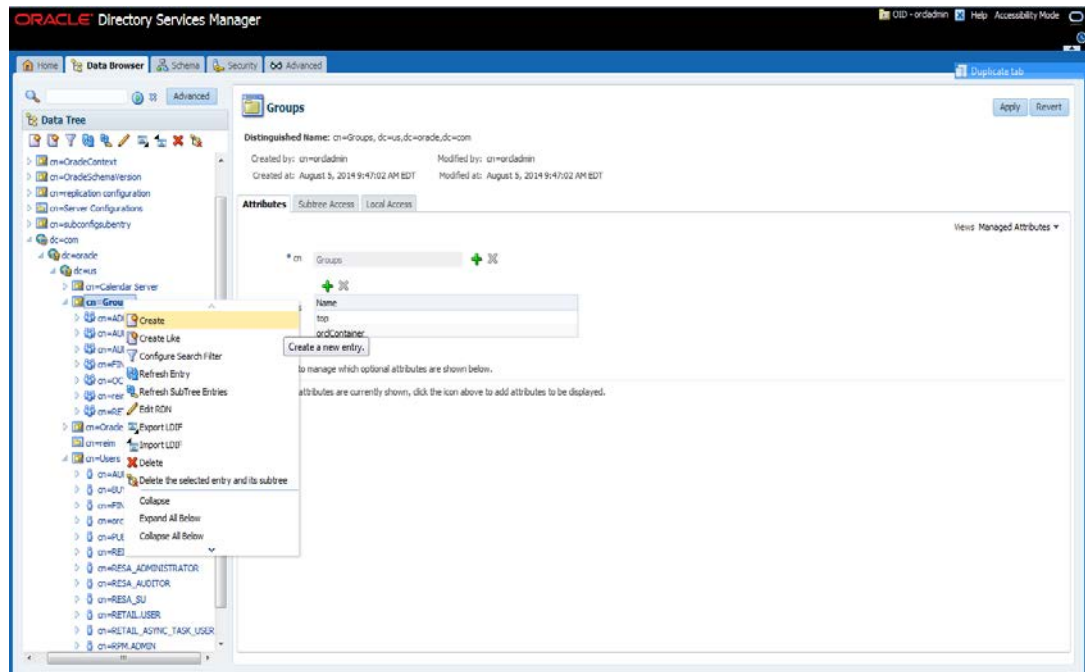
- c. From the OID Connect dialog, click the **Connect** button.
- d. From the Oracle Internet Directory Welcome Screen, select the Data Browser tab. The DataBrowser tree shows how to find the “cn=Group” element.



- e. From the Data Tree panel of the ODSM screen, navigate to dc=com,dc=oracle,dc=us,cn=Groups.

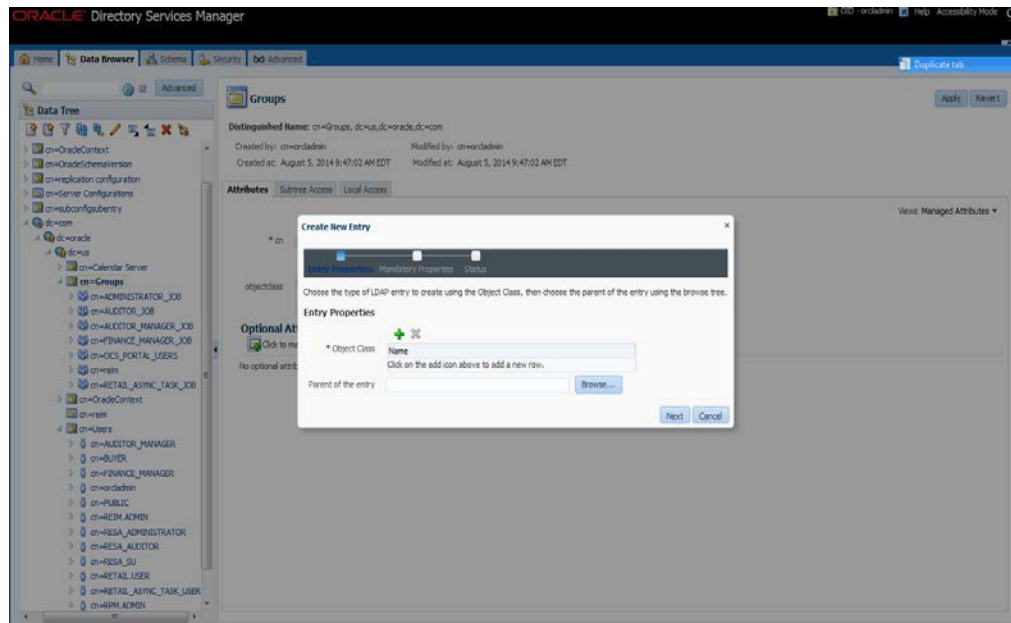


f. Right-click `cn=Groups` and select **Create**.

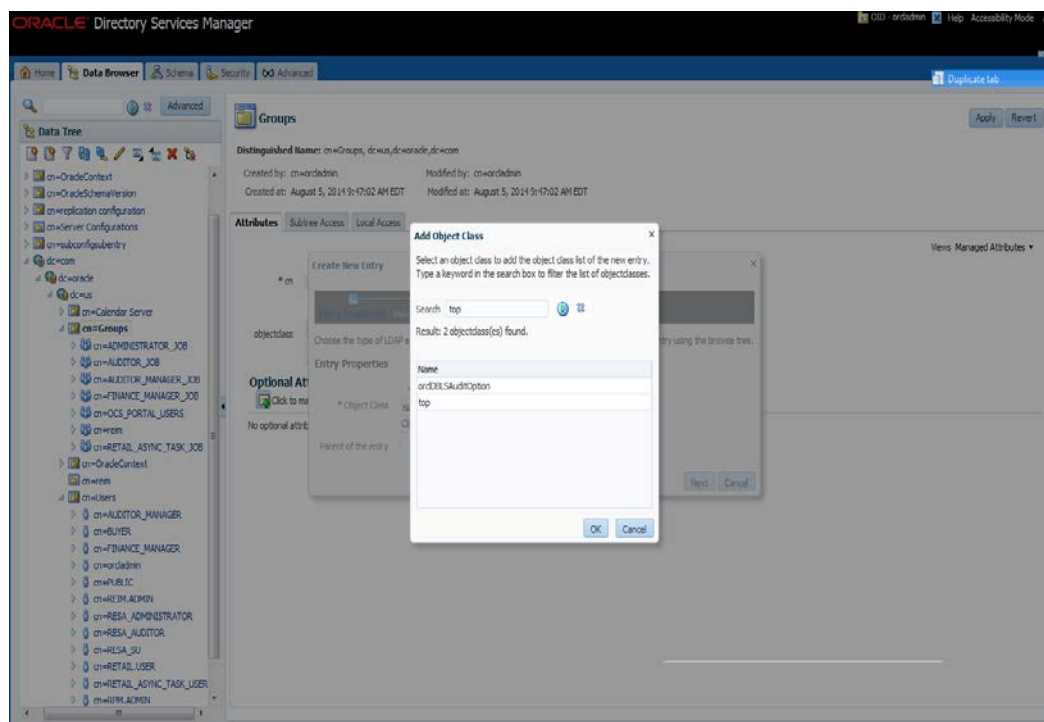


g. From the Create New Entry dialog, click the + icon and add the following Object Classes:

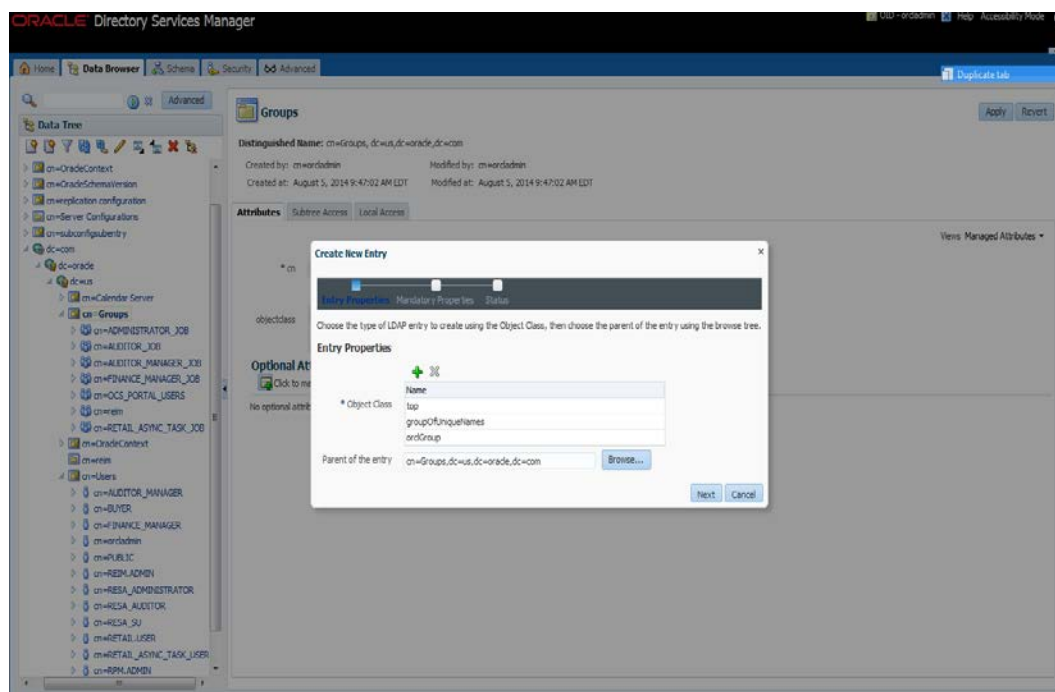
- top
- groupOfUniqueNames
- orclGroup



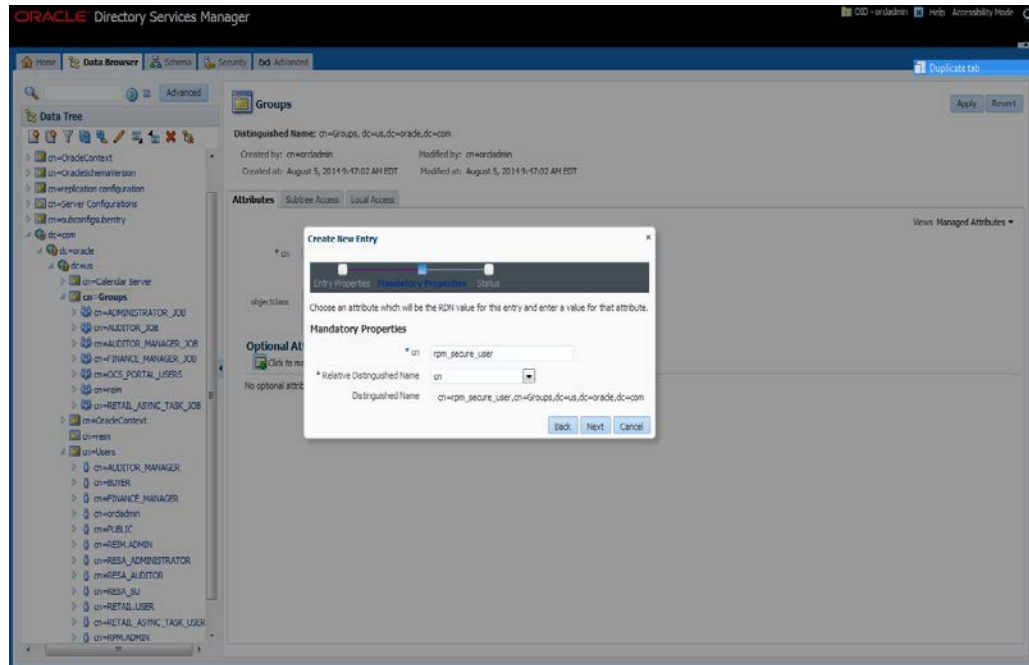
- h. From the Add Object Class drop down menu select **top**.



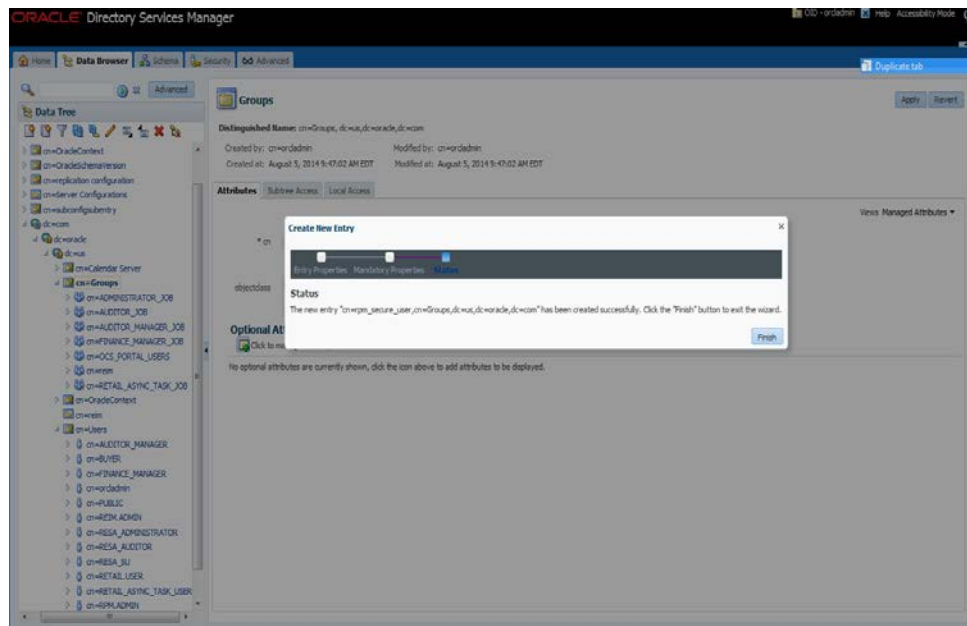
- i. When finished adding all the object classes the screen will look like as follows:
Enter: cn=Groups,dc=us,dc=oracle,dc=com into the Parent of the entry field.
Click Next



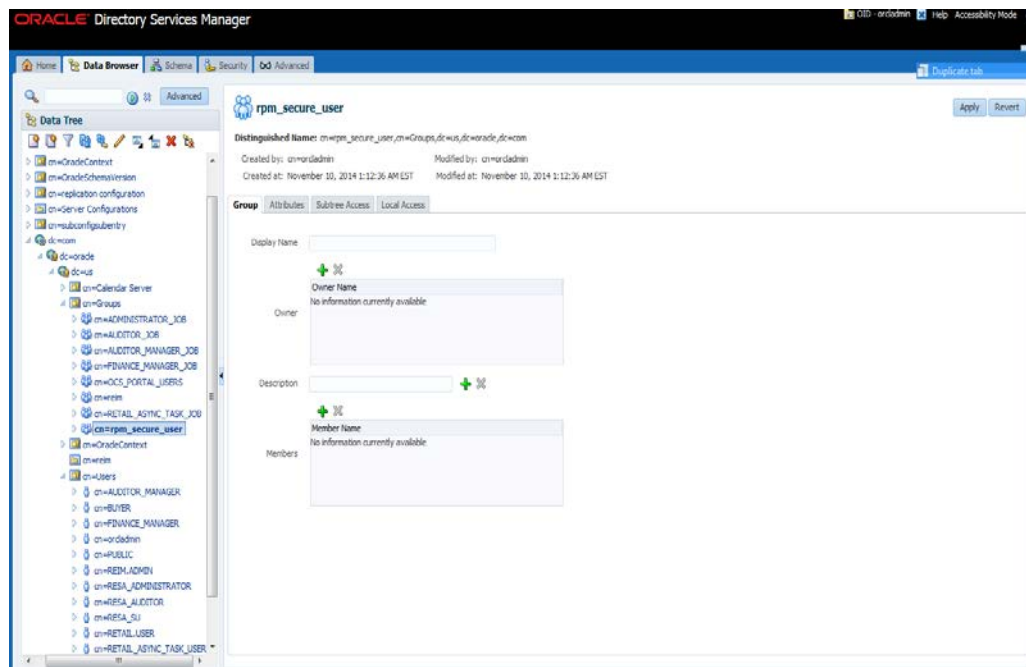
- j. On the “*cn” text field enter: “rpm_secure_users”. On Relative Distinguished Name field enter: cn and click **Next**.



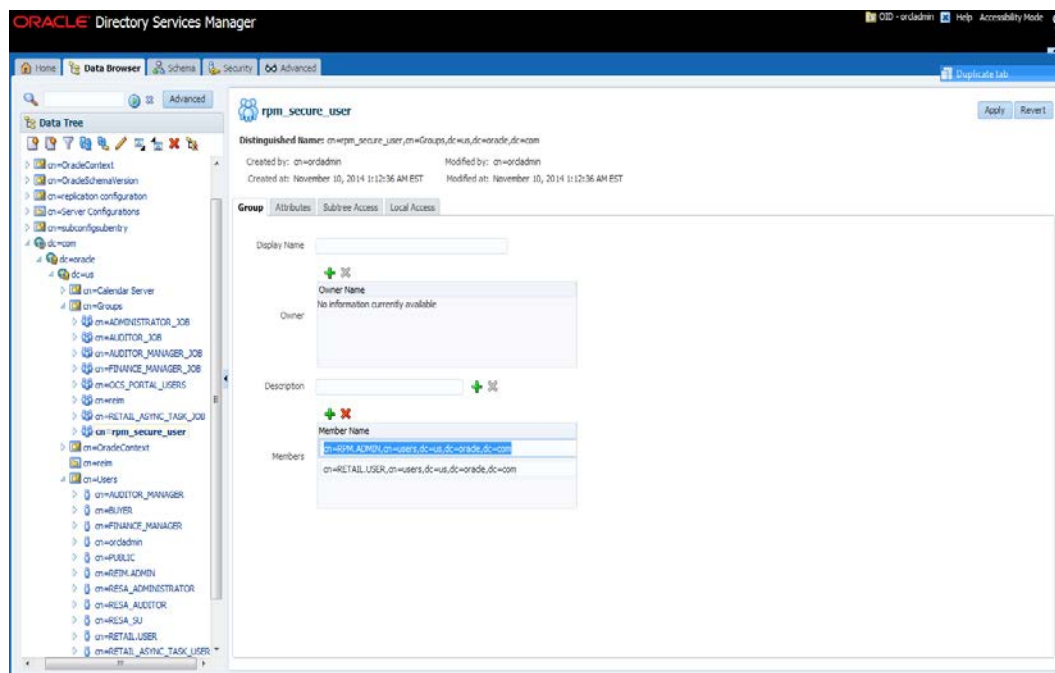
- k. Click **Finish**.



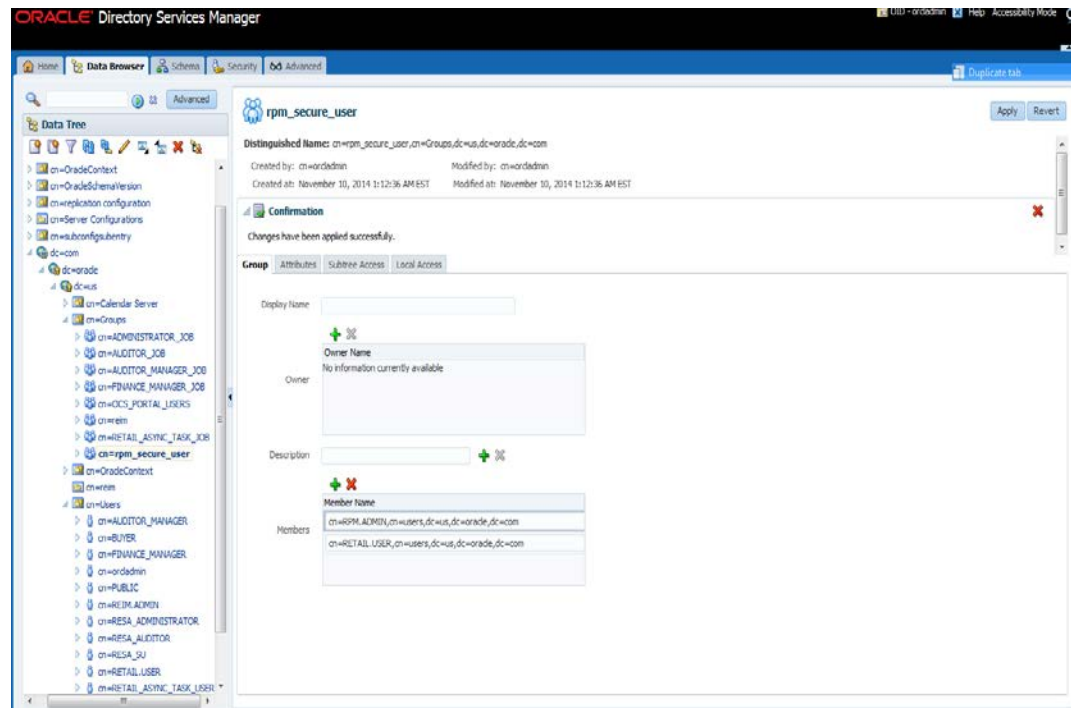
After applying the changes, your screen should look similar to the following when clicking on the rpm_secure_users group:



- I. Click on the + Members button and add the following users and click Apply:
 - i. cn=retail.user,cn=users,dc=us,dc=oracle,dc=com
 - ii. cn=rpm.admin,cn=users,dc=us,dc=oracle,dc=com

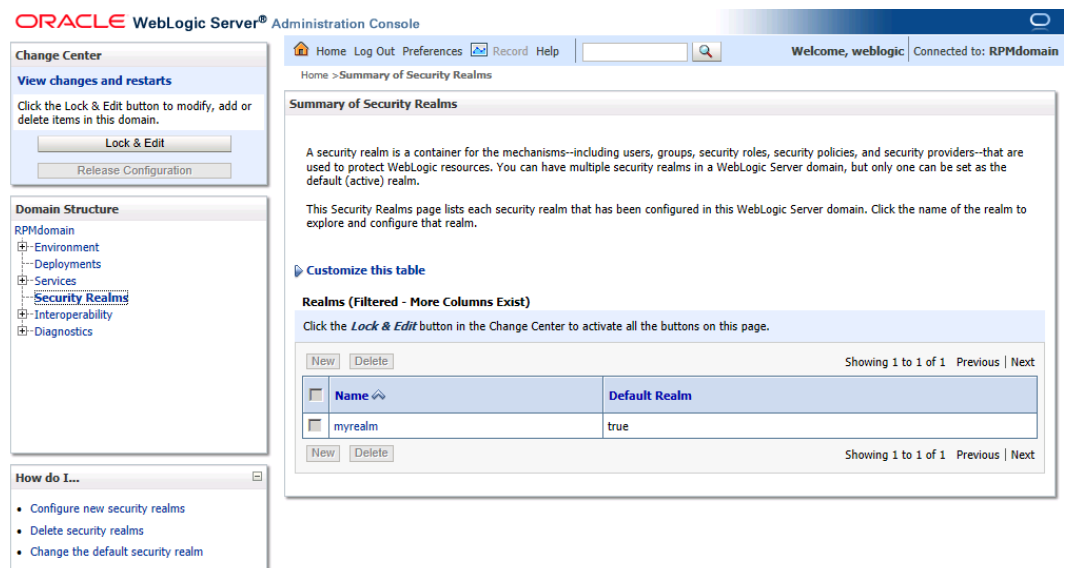


m. Click **Apply**. The following screen is displayed:



Create OID Authentication Provider

1. Log in to the Administration Console.
<http://<host>:<port>/console/>
2. In the Domain Structure frame, click **Security Realms**.



3. In the Realms table, click **myrealm**. The Settings for myrealm page is displayed.

ORACLE WebLogic Server® Administration Console

Welcome, weblogic Connected to: RPPdomain

Home > Summary of Security Realms > myrealm

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

General RDBMS Security Store User Logout Performance

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Save

Use this page to configure the general behavior of this security realm.

Note: If you are implementing security using JACC (Java Authorization Contract for Containers as defined in JSR 115), you must use the DD Only security model. Other WebLogic Server models are not available and the security functions for Web applications and EJBs in the Administration Console are disabled.

Name: myrealm The name of this security realm. [More Info...](#)

Security Model Default: DD Only Specifies the default security model for Web applications or EJBs that are secured by this security realm. You can override this default during deployment. [More Info...](#)

☒ Combined Role Mapping Enabled Determines how the role mappings in the Enterprise Application, Web application, and EJB containers interact. This setting is valid only for Web applications and EJBs that use the Advanced security model and that initialize roles from deployment descriptors. [More Info...](#)

☒ Use Authorization Providers to Protect JMX Access Configures the WebLogic Server MBean servers to use the security realm's Authorization providers to determine whether a JMX client has permission to access an MBean attribute or invoke an MBean operation. [More Info...](#)

Advanced

Save

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

WebLogic Server Version: 10.3.6.0
Copyright © 1996, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

4. Click the Providers tab.

ORACLE WebLogic Server® Administration Console

Welcome, weblogic Connected to: RPPdomain

Home > Summary of Security Realms > myrealm > Providers

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path KeyStores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

New Delete Reorder

Name	Description	Version
DefaultAuthenticator	WebLogic Authentication Provider	1.0
DefaultIdentityAssertion	WebLogic Identity Assertion provider	1.0

New Delete Reorder

Showing 1 to 2 of 2 Previous Next

5. Click **Lock & Edit** and then click **New**. The Create a New Authentication Provider page is displayed.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Home > Summary of Security Realms > myrealm > Providers > Summary of Security Realms > myrealm > Providers

Create a New Authentication Provider

OK Cancel

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.
* Indicates required fields

The name of the authentication provider.

* Name:

This is the type of authentication provider you wish to create.

Type:

OK 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

RPMdomain

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

How do I...

- Manage security providers
- Configure Authentication and Identity Assertion providers

System Status

Health of Running Servers

Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)

WebLogic Server Version: 10.3.6.0
Copyright © 1996, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

6. Enter **OIDAuthenticator** in the Name field and select **OracleInternetDirectoryAuthenticator** as the type.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Home > Summary of Security Realms > myrealm > Providers > Summary of Security Realms > myrealm > Providers

Create a New Authentication Provider

OK Cancel

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.
* Indicates required fields

The name of the authentication provider.

* Name:

This is the type of authentication provider you wish to create.

Type:

OK 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

RPMdomain

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

How do I...

- Manage security providers
- Configure Authentication and Identity Assertion providers

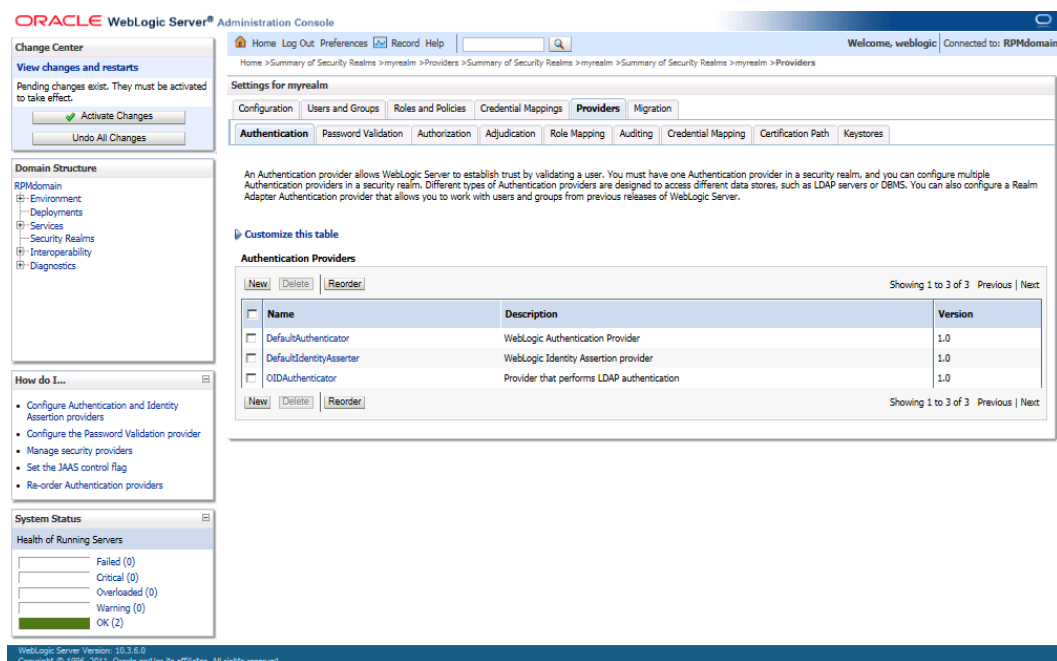
System Status

Health of Running Servers

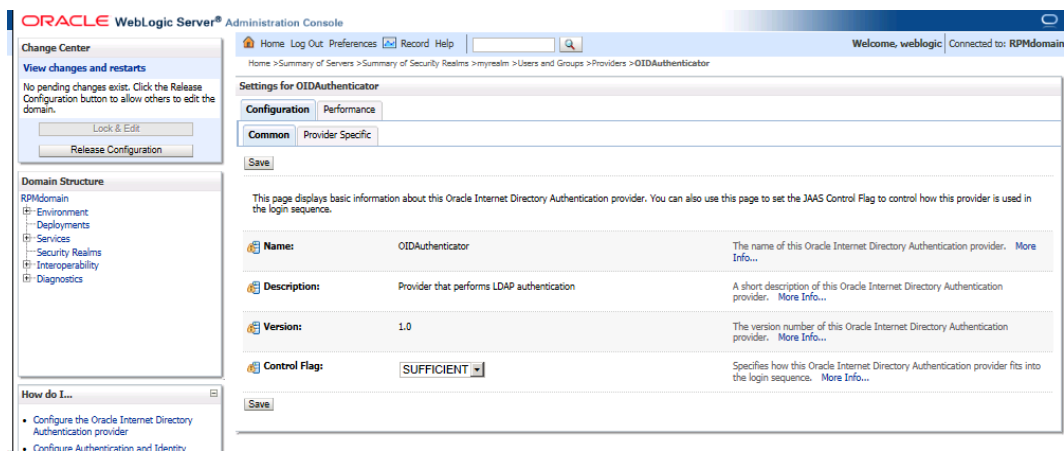
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)

WebLogic Server Version: 10.3.6.0
Copyright © 1996, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

7. Click **OK**. The OID Provider will now be visible on the Providers tab.



8. Click on the **OIDAuthenticator**. The authenticator configuration screen will appear.



9. Set the Control Flag field to **SUFFICIENT** and click **Save**.

10. Click the **Provider Specific** tab.

11. Supply your LDAP connection and credentials.

The entries below are examples only. You should match the entries to your OID

- Host: <OID Server name>
- Port: <OID port> (Example: 3060 or 389)
- Principal: <cn=orcladmin> (provide the OID admin user)
- Credential: <password> (provide the password of cn=orcladmin)
- User Base DN: (Example: cn=Users,dc=us,dc=oracle,dc=com)
- Group Base DN: (Example: cn=Groups,dc=us,dc=oracle,dc=com)

- All Users Filter : (&(cn=*)(objectclass=person))

View changes and restarts
Pending changes exist. They must be activated to take effect.

[Activate Changes](#)
[Undo All Changes](#)

Domain Structure
MacDomain
Environment
Servers
Clusters
Virtual Hosts
Migratable Targets
Coherence Servers
Coherence Clusters
Machines
Work Managers
Startup and Shutdown Classes
Deployments
Services
Security Extens...

How do I...?

- Configure the Oracle Internet Directory Authentication provider
- Configure Authentication and Identity Assertion providers
- Manage security providers

System Status
Health of Running Servers
Failed (0)

Settings for OIAuthenticator
Configuration Performance
Common **Provider Specific**

[Save](#)

Use this page to define the provider specific configuration for this Oracle Internet Directory Authentication provider.

Connection

Host: <OIDHostname> The host name or IP address of the LDAP server. [More Info...](#)

Port: 3060 The port number on which the LDAP server is listening. [More Info...](#)

Principal: cn=orcladmin The Distinguished Name (DN) of the LDAP user that WebLogic Server should use to connect to the LDAP server. [More Info...](#)

Credential: ***** The credential (usually a password) used to connect to the LDAP server. [More Info...](#)

Confirm Credential: *****

☐ **SSL Enabled** Specifies whether the SSL protocol should be used when connecting to the LDAP server. [More Info...](#)

Users

User Base Dn: dc=us,dc=oracle,dc=com The base distinguished name (DN) of the tree in the LDAP directory that contains users. [More Info...](#)

All Users Filter: (&(cn=*)(objectclass=per) An LDAP search filter for finding all users beneath the base user distinguished name (DN). Note: If you change the user name attribute to a type other than cn, you must also change the user object class to the correct Oracle Internet Directory object class. [More Info...](#)

12. Click Save.

View changes and restarts
Pending changes exist. They must be activated to take effect.

[Activate Changes](#)
[Undo All Changes](#)

Domain Structure
MacDomain
Environment
Servers
Clusters
Virtual Hosts
Migratable Targets
Coherence Servers
Coherence Clusters
Machines
Work Managers
Startup and Shutdown Classes
Deployments
Services
Security Extens...

How do I...?

- Configure the Oracle Internet Directory Authentication provider
- Configure Authentication and Identity Assertion providers
- Manage security providers

System Status
Health of Running Servers
Failed (0)

Settings for OIAuthenticator
Configuration Performance
Common **Provider Specific**

[Save](#)

Use this page to define the provider specific configuration for this Oracle Internet Directory Authentication provider.

Connection

Host: <OIDHostname> The host name or IP address of the LDAP server. [More Info...](#)

Port: 3060 The port number on which the LDAP server is listening. [More Info...](#)

Principal: cn=orcladmin The Distinguished Name (DN) of the LDAP user that WebLogic Server should use to connect to the LDAP server. [More Info...](#)

Credential: ***** The credential (usually a password) used to connect to the LDAP server. [More Info...](#)

Confirm Credential: *****

☐ **SSL Enabled** Specifies whether the SSL protocol should be used when connecting to the LDAP server. [More Info...](#)

Users

User Base Dn: dc=us,dc=oracle,dc=com The base distinguished name (DN) of the tree in the LDAP directory that contains users. [More Info...](#)

All Users Filter: (&(cn=*)(objectclass=per) An LDAP search filter for finding all users beneath the base user distinguished name (DN). Note: If you change the user name attribute to a type other than cn, you must also change the user object class to the correct Oracle Internet Directory object class. [More Info...](#)

13. Navigate to Security Realms – myrealm – and then click the Providers tab.

The screenshot shows the Oracle WebLogic Server Administration Console. The left sidebar contains the 'Change Center' with 'View changes and restarts' and 'Activate Changes' buttons. Below it is the 'Domain Structure' tree with 'RPMDomain' expanded, showing 'Environment', 'Deployments', 'Services', 'Security Realms', 'Interoperability', and 'Diagnostics'. The 'How do I...' section lists tasks like 'Configure Authentication and Identity Assertion providers'. The 'System Status' section shows 'Health of Running Servers' with 'Failed (0)', 'Critical (0)', 'Overloaded (0)', 'Warning (0)', and 'OK (2)'.

The main content area shows the 'Providers' tab for the 'myrealm' security realm. It includes a breadcrumb trail: 'Home > Providers > Summary of Security Realms > myrealm > Summary of Security Realms > myrealm > Providers > Providers'. The 'Providers' tab is selected, showing a table of authentication providers:

Name	Description	Version
DefaultAuthenticator	WebLogic Authentication Provider	1.0
DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0
OIDAuthenticator	Provider that performs LDAP authentication	1.0

Buttons for 'New', 'Delete', and 'Reorder' are available. The status bar at the bottom indicates 'WebLogic Server Version: 10.3.6.0'.

14. Click Reorder.

15. Order OIDAuthenticator first and DefaultAuthenticator second.

The screenshot shows the 'Reorder Authentication Providers' dialog box. The left sidebar is the same as in the previous screenshot. The main content area shows the 'Reorder Authentication Providers' dialog with the following text:

You can reorder your Authentication Providers using the list below. By reordering Authentication Providers, you can alter the authentication sequence.

Select authenticator(s) in the list and use arrows to move them up and down in the list.

Authentication Providers:

Available:

- ☒ OIDAuthenticator
- ☐ DefaultAuthenticator
- ☐ DefaultIdentityAsserter

Buttons for 'OK' and 'Cancel' are at the bottom.

16. Click Ok.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Home > Providers > Summary of Security Realms > myrealm > Summary of Security Realms > myrealm > Providers > Summary of Security Realms > myrealm > Providers

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder Showing 1 to 3 of 3 Previous Next

Name	Description	Version
<input type="checkbox"/> OIDAuthenticator	Provider that performs LDAP authentication	1.0
<input type="checkbox"/> DefaultAuthenticator	WebLogic Authentication Provider	1.0
<input type="checkbox"/> DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0

New Delete Reorder Showing 1 to 3 of 3 Previous Next

WebLogic Server Version: 10.3.6.0

17. Once your changes are saved, click **Activate Changes**.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RPMdomain

Home > Providers > Summary of Security Realms > myrealm > Summary of Security Realms > myrealm > Providers > Summary of Security Realms > myrealm > Providers

Messages

✓ All changes have been activated. However 2 items must be restarted for the changes to take effect.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

New Delete Reorder Showing 1 to 3 of 3 Previous Next

Name	Description	Version
<input type="checkbox"/> OIDAuthenticator	Provider that performs LDAP authentication	1.0
<input type="checkbox"/> DefaultAuthenticator	WebLogic Authentication Provider	1.0
<input type="checkbox"/> DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0

New Delete Reorder Showing 1 to 3 of 3 Previous Next

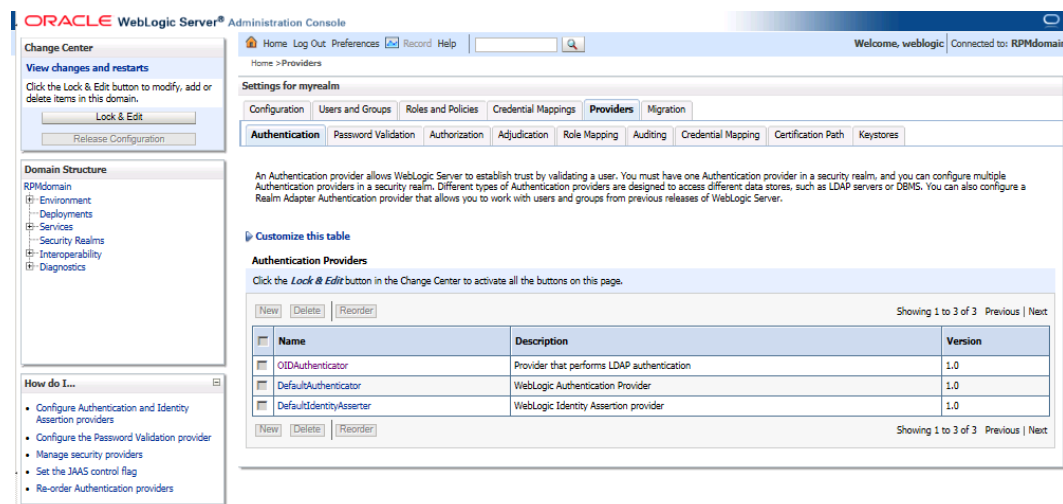
WebLogic Server Version: 10.3.6.0

18. Shut down all servers and restart the admin server.

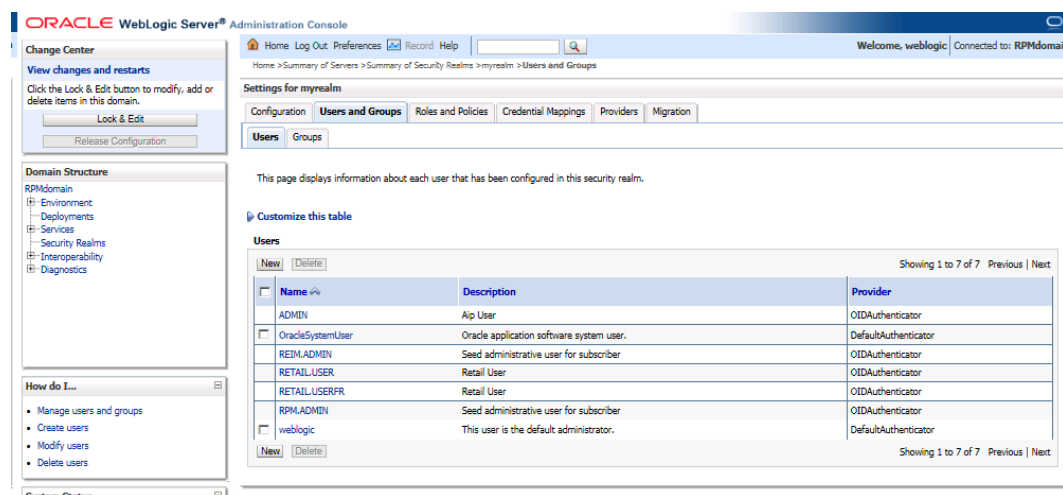
Verify and Set OID Authenticator

- Log in to the Administration Console.
http://<host>:<port>/console/
- In the Domain Structure frame, click Security Realms.

3. In the Realms table, click Default Realm Name. The Settings page is displayed.
4. Click the Providers tab. You must see the OID Provider in that list.



5. Click the Users and Groups tab to see a list of users and groups contained in the configured authentication providers.



Expand the RPM Application Distribution

To expand the RPM application distribution, do the following.

1. Log into the UNIX server as the user who owns the WebLogic installation. Create a new staging directory for the RPM application distribution (rpm14application.zip). There should be a minimum of 2 GB disk space available for the application installation files.

Example: /u00/webadmin/media/rpm

This location is referred to as STAGING_DIR for the remainder of this chapter.

2. Copy rpm14application.zip to STAGING_DIR and extract its contents.

Clustered Installations – Preinstallation Steps

Skip this section if you are not clustering the application server.

If you are installing the RPM application to a clustered WebLogic Application Server environment, there are some extra steps you need to take before running the RPM application installer. In these instructions, the application server node with the ORACLE_HOME you used for the RPM installer is referred to as the *master node*. All other nodes are referred to as the *remote nodes*.

1. Before starting the RPM Application Installer, make sure that you are able to start and stop the managed servers that are part of the RPM Application Cluster from the WebLogic Administration Console.
2. When the RPM Application Installer displays the screen in which it asks for the information related to the JMS Provider, we recommend entering these values:

- 3. Insert into all remote nodes
\$WLS_HOME/wlserver_10.3/server/lib/weblogic.policy file changes, the same RPM entries for java security permissions you entered on the master node. See the “[Start the Managed Servers](#)” section for additional information.

Provide the Hibernate4 Jar File

The RPM application requires hibernate4 jar files to be installed. These files should be downloaded from <http://www.hibernate.org>.

Extract the required Hibernate 4.3.5 jar files and place them within the application servers STAGING_DIR/rpm/application/hibernate4 directory before running the installer. The installer will then install the jar files within the application for you.

The required jars are as follows:

- hibernate-commons-annotations-4.0.*.jar
- hibernate-core-4.3.*.jar
- hibernate-ehcache-4.3.*.jar
- hibernate-jpa-2.1-api-1.0.*.jar
- jboss-logging-3.1.*.jar
- jboss-transaction-api_1.2_spec-1.0.*.jar

The required jar files are located in the <HIBERNATE_EXTRACT_DIR>\hibernate-release-4.3.5.Final.tgz\hibernate-release-4.3.5.Final.tar\hibernate-release-4.3.5.Final\lib\required\directory.

Run the RPM Application Installer

Once you have a WebLogic instance that is configured and started, you can run the RPM application installer. This installer configures and deploys the RPM application and Java WebStart client files.

Note: See [Appendix: RPM Application Installer Screens](#) for details on every screen and field in the application installer. The screenshots contain instructions that are necessary to result in a working application.

1. Change directories to STAGING_DIR/rpm/application.
2. Set the ORACLE_HOME, WEBLOGIC_DOMAIN_HOME and JAVA_HOME environment variables. ORACLE_HOME should point to your WebLogic installation. JAVA_HOME should point to the Java 7.0 (1.7.) JDK. WEBLOGIC_DOMAIN_HOME should point to your WebLogic domain.
3. If a secured datasource is going to be configured you also need to set "ANT_OPTS" so the installer can access the key and trust store that is used for the datasource security:

```
export ANT_OPTS="-Djavax.net.ssl.keyStore=<PATH TO KEY STORE> -  
Djavax.net.ssl.keyStoreType=jks -Djavax.net.ssl.keyStorePassword=<KEYSTORE  
PASSWORD> -Djavax.net.ssl.trustStore=<PATH TO TRUST STORE> -  
Djavax.net.ssl.trustStoreType=jks -  
Djavax.net.ssl.trustStorePassword=<TRUSTSTORE PASSWORD>"
```

An example of this would be:

- ```
export ANT_OPTS="-
Djavax.net.ssl.keyStore=/u00/webadmin/product/wls_retail
/wlserver_10.3/server/lib/msp52278.keystore -Djavax.net.ssl.keyStoreType=jks -
Djavax.net.ssl.keyStorePassword=retail123 -Djavax.net.ssl.trustStore=/
u00/webadmin/product/wls_retail
/wlserver_10.3/server/lib/msp2278.keystore -Djavax.net.ssl.trustStoreType=jks
-Djavax.net.ssl.trustStorePassword=retail123"
```
4. If you are using an X server such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
  5. Run the install.sh script. This launches the installer. After installation is complete, a detailed installation log file is created (rpm14install.<timestamp>.log).

---

**Note:** The values you enter in the installer screen, "Setup Application Users," have specific requirements for RPM to work properly. See the screen description in [Appendix: RPM Application Installer Screens](#) for more details. The screenshots contain instructions that are necessary to result in a working application.

---

## Resolving Errors Encountered During Application Installation

If the application installer encounters any errors, it halts execution immediately. You can run the installer in silent mode so that you do not have to retype the settings for your environment. See [Appendix: Installer Silent Mode](#) in this document for instructions on silent mode.

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

Because the application installation is a full installation every time, any previous partial installations are overwritten by the successful installation.

## Clustered Installations – Post-Installation Steps

If you are installing the RPM application to a clustered WebLogic Server environment, there are some extra steps you need to take to complete the installation. In these instructions, the application server with the ORACLE\_HOME you used for the RPM installer is referred to as the master server. All other nodes are referred to as the remote servers.

1. The RPM batch files should be copied from the master node to each of the remote nodes under the same path as on the master node. You should take the `$WEBLOGIC_DOMAIN_HOME/retail/<rpmdir>/rpm-batch` directory and copy it onto the remote nodes under the same path.
2. For retailers who install batch on either node of the cluster, `launchRpmBatch.sh` script should be modified on each remote node to point to the local RPM instance. The RPM URL is set in the `PROVIDER_URL` variable. This script is located at `$WEBLOGIC_DOMAIN_HOME/retail/<rpmdir>/rpm-batch/scripts/launchRpmBatch.sh`.
3. The Oracle Retail Installation creates some security files on `$WEBLOGIC_DOMAIN_HOME/retail/<rpm_application_name>/config` directory. Copy this directory to each remote node of the Cluster, matching the full path of the location of this directory on main node.
4. The Oracle Retail Installation creates some properties files on `$WEBLOGIC_DOMAIN_HOME/retail/<rpm_application_name>/properties` directory. Copy this directory to each remote node of the Cluster, matching the full path of the location of this directory on main node.

## Review and/or Configure Oracle Single Sign-On

---

**Note:** This step is only needed if you plan on setting up the RPM application using Single Sign On (SSO) authentication. This can be skipped if SSO is not going to be configured for this environment. The Oracle Access manager must be configured and the Oracle http server (Webtier and webgate) must be registered into the Oracle Access Manager.

---

### Create the RPM SSO provider in the RPMdomain:

1. Shut down all the servers of the Weblogic Domain created.
2. Once you copy the contents to `<INSTALL_DIR>` copy the `rpm14-security.zip` present in `<INSTALL_DIR>/rpm/application/rpm14` to the `WEBLOGIC_DOMAIN_HOME/lib` and extract its contents in the folder.
3. Start the domain admin server.
4. Log into the WebLogic console.
5. Navigate to: security realms -> myrealm (default realm) -> providers.
6. Start a Lock and Edit session.
7. Click New provider.
8. Select the provider type from the list: `RpmWlsSsoAuthenticator`.
9. Set the provider name (Default: `RpmSsoAuthenticator`).
10. Click **Ok**.
11. Open the new provider configuration.
12. Under Common, set the Control Flag to `SUFFICIENT`.
13. Click **Provider Specific**.
14. Check that the `GroupName` is set to the name of the group used for RPM secure users (`rpm_secure_users` by default).
15. All other values under the Provider Specific tab can be left as the default value.
16. Click **Ok**.

17. On the provider list, click **Reorder**.
18. Move the RpmWlsSsoAuthenticator to the top of the list, or above the DefaultAuthenticator.
19. Click **Ok**.
20. Click **Activate Changes**.
21. Shutdown the domain.
22. Start the admin and managed servers for the domain.

After the SSO provider is created in the RPMdomain, you will also have to set the protection of the RPM application resources correctly in the Application Domain that has been registered in the Oracle Access Manager.

In the Webtier/Webgate http server you need to set the mod\_wl\_ohs.conf file to redirect the http call to the where the RPM application has been deployed.

For example, in mod\_wl\_ohs.conf set:

```
<Location /rpm-client >
 WebLogicCluster msp52278.us.oracle.com:17011
 SetHandler weblogic-handler
 ErrorPage downtime.html
</Location>
```

Then in Oracle Access Manager, set the protection of the resources in the Application Domain that has been registered for the RPM application. You must protect the /rpm-client/launch resource and unprotect the rest:

Resource URL: / rpm-client/launch

Protection Level: Protected

Authentication Policy: Protected Resource Policy

Authorization Policy: Protected Resource Policy

Resource URL: / rpm-client/.../\*

Protection Level: Excluded

## Sign the RPM Client Configuration Jar File

There is some client-side configuration that the installer performs which results in a modified rpm\_client\_config.jar file after installation. Because of this, the jar file cannot be pre-signed by Oracle. The installer now provides an option to sign the jar by asking some details but if decide not to do it using the installer, the user must sign this jar file after the installer has completed.

To create an example key called foo, the following command can be run:

```
$JAVA_HOME/bin/keytool -genkey -alias foo
```

This command prompts you for a keystore password along with organizational info.

Once complete, the keystore alias resides in the default location in the user's home directory (for example, ~/.keystore). If you get an error message saying that the keystore has been tampered with, try renaming or deleting the ~/.keystore file and running the keytool command again.

The rpm\_client\_config.jar file is located in \$WEBLOGIC\_DOMAIN\_HOME/servers/<rpm-managedserver>/tmp/\_WL\_user/rpm/<38o5n1 >/war/lib. To sign the rpm\_client\_config.jar file using your alias and keystore, run the jarsigner utility.



---

**Example:** jarsigner  
 \$WEBLOGIC\_DOMAIN\_HOME/servers/rpm-  
 managedserver/tmp/\_WL\_user/rpm/38o5n1 /war/  
 lib/rpm\_client\_config.jar foo

---

If you are clustering the application server you need to copy the signed rpm\_client\_config.jar file to the same path under \$ORACLE\_HOME on all remote nodes. Consult the **jarsigner** documentation from Sun for further information on the JAR signing process.

You also need to sign in same jar file inside WebLaunchServlet.war which can be found in the rpm14.ear location in the stage directory of the managed server. Copy the ear file in a temporary location extract the ear files sign the jar file and compress them again and replace it with the one in the staging directory. The above is needed to avoid unsigned entries in jar after restarting the server in the future. Once you restart the weblogic server, files will re-loaded from the once in stage directory. Hence these steps are needed. For signing this jar, the user must have a certificate.

## Transaction Timeout

This section describes how to establish settings for a transaction timeout. A transaction timeout is the maximum duration, in seconds, for transactions on the application server. Any transaction that is not required to complete before this timeout is rolled back. To set up transaction timeouts, complete these steps:

1. Log in to the WebLogic Server 10.3.6 Administration Console.
2. Click on the Domain link.
3. Under Configuration, click **JTA**.
4. Click **Lock and Edit**.
5. Set the Timeout Seconds (for example, 600 seconds).
6. Click **Activate Changes**.

## Backups Created by Installer

The RPM application installer backs up previous batch, JMS bindings, and WebStart client installations by renaming them with <timestamp> suffixes. This is done to prevent the removal of any custom changes you might have. These backup directories can be safely removed without affecting the current installation.

---

**Examples:** rpm-batch.200605011726, sbynjndi.200605011726,  
 rpm.200605011726

---

## Test the RPM Application

After the application installer finishes, a working RPM application installation should result, if the users were created properly.

For LDAP authentication, the application will not log you in properly unless you have a row for the users in question in the database on the rsm\_user\_role table. The following is an example of how to add rows if they have not been added.

```

insert into rsm_user_role
(id, user_id, role_id, start_date_time, end_date_time)
select rsm_user_role_seq.nextval,
 'retail.user',
 -1001,
 nvl(get_vdate,sysdate) - 365,
 null
from dual;

```

If problems occur when trying to start the RPM application, ensure proxies are turned off.

To launch the application client, open a Web browser and access the JnlpLaunchServlet, naming the RPM JNLP template file (rpm\_jnlp\_template.vm).

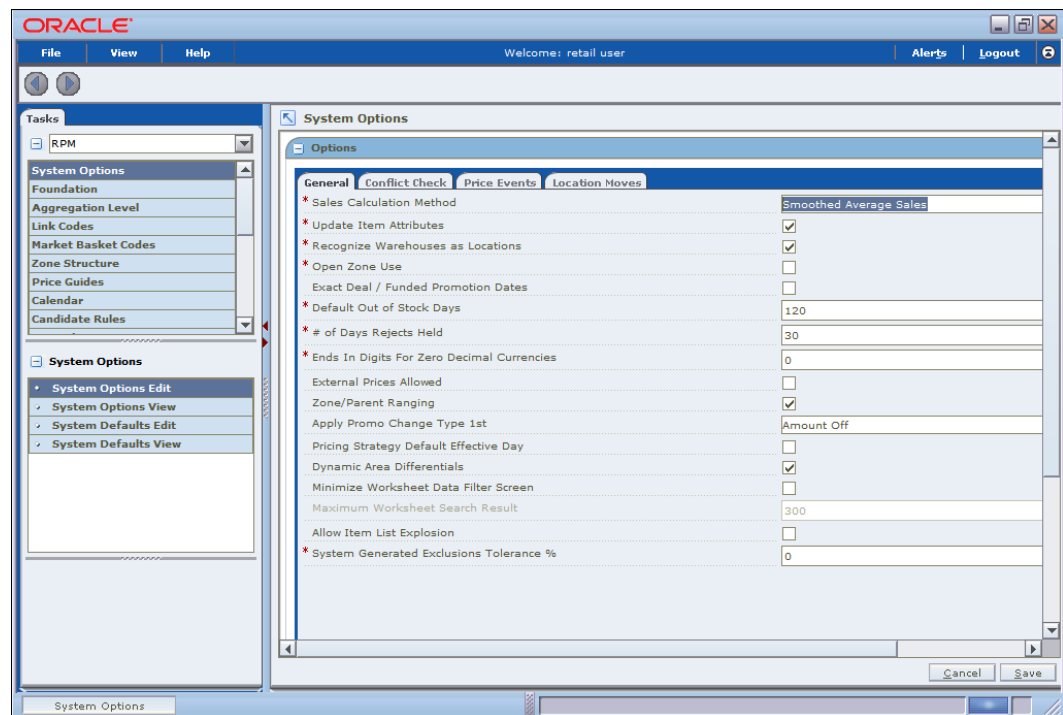
---

**Example:** [http://appserver1:17011/rpm-client/launch?template=rpm\\_jnlp\\_template.vm](http://appserver1:17011/rpm-client/launch?template=rpm_jnlp_template.vm)

---

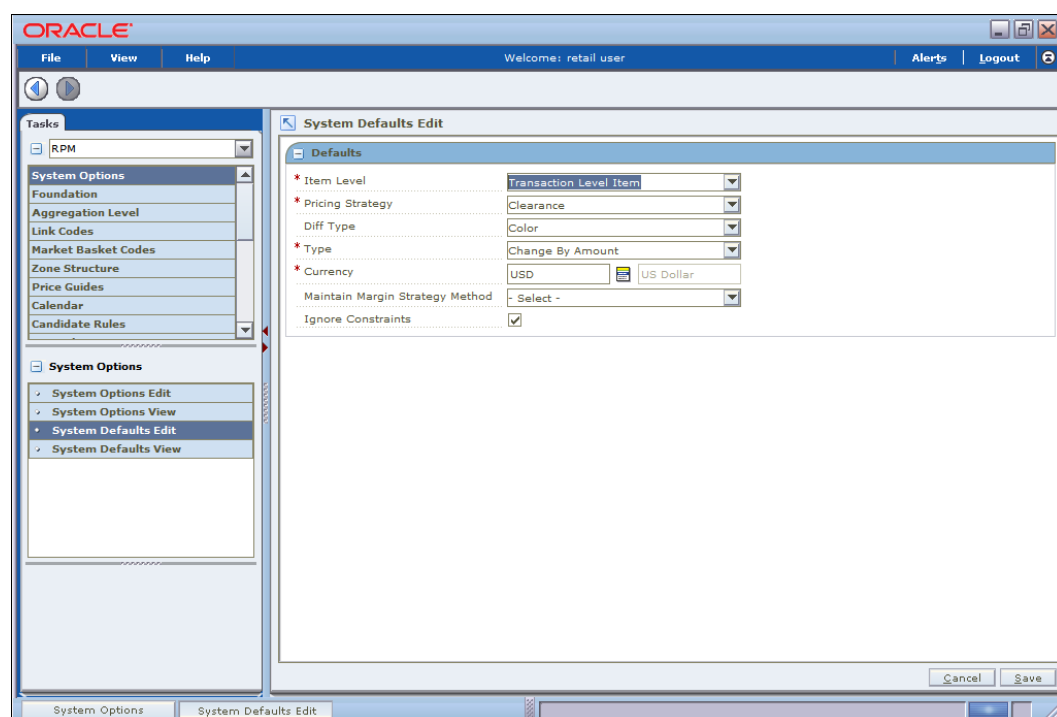
When you are in the RPM application, do the following to add a rpm\_system\_options row required by RPM for system use.

1. On the left side of the screen, select System Options.
2. Select **System Options Edit**.
3. In the **lower right part of the screen**, click **Save**.



4. To add an rpm\_system\_options\_def row required by RPM needs for system use, do the following.
5. Select System Defaults Edit.

6. In the lower right part of the screen, click **Save**.



## RPM Batch Scripts

The RPM application installer configures and installs the batch scripts under `<retail_home>/rpm-batch.` You will run the RPM java batch pgms with a java wallet alias (for example, RETAIL.USER1) that you created in the installer screens. The following is an example execution of a RPM java batch script.

```
./<RPMbatchscriptname>.sh RETAIL.USER1
```

---

**Note:** Make sure that JAVA\_HOME is set to the appropriate Java JDK (the same JDK that has been used by WebLogic Server) and ORACLE\_HOME is set to weblogic installation before running the RPM batch programs.

---

## RPM Batch Scripts that call sqlplus (plsql batch)

In some RPM batch scripts sqlplus is called, so a profile should be set up for this user. A prerequisite for this would be Oracle database or Oracle client installed on the server. The below example assumes that a batch user rpmbatch was created in the Oracle Wallet (different from the Java wallet) and added to the tnsnames.ora, as explained in [Appendix: Setting Up Password Stores with Oracle Wallet](#).

The batch scripts calling sqlplus are as follows:

- clearancePriceChangePublishExport.sh
- priceEventItemListPurgeBatch.sh
- priceEventPayloadPopulationBatch.sh
- primaryZoneModificationsBatch.sh
- promotionPriceChangePublishExport.sh
- purgeExpiredExecutedOrApprovedClearancesBatch.sh
- purgePayloadsBatch.sh
- purgeUnusedAndAbandonedClearancesBatch.sh
- regularPriceChangePublishExport.sh
- RPMtoORPOSPublishBatch.sh
- RPMtoORPOSPublishExport.sh

Example profile.sh

```
#!/bin/sh

#Need the Oracle Home set to aim at ORACLE Client or db on the server RPM # is
installed on
ORACLE_HOME=/u00/oracle/product/12.1.0.2

#Java Home for the Oracle install
JAVA_HOME=$ORACLE_HOME/jdk

#Add the Oracle and Java bin's to path
PATH=$ORACLE_HOME/bin:$JAVA_HOME/bin:$PATH

export PATH ORACLE_HOME JAVA_HOME

#Path to directory with tnsnames.ora, ewallet.pl2, cwallet.sso &
#sqlnet.ora (You will build these files as explained in Appendix E Setting #Up
Password Stores with Oracle Wallet)
TNS_ADMIN=/u00/webadmin/product/10.3.x/WLS/user_projects/domains/APPDomain/ret
ail/rpm14/config/wallet
export TNS_ADMIN
echo "ORACLE_HOME=${ORACLE_HOME}"
echo "JAVA_HOME=${JAVA_HOME}"
echo "PATH=${PATH}"
```

To source the profile above, do the following:

```
$. ./profile.sh
```

While running the plsqli batch script the connect string as follows (/@rpmbatch that you created using the instructions in ["Appendix: Setting Up Password Stores with Oracle Wallet."](#)

```
./RPMtoORPOSPublishExport.sh /@rpmbatch 0 log error
```

## Online Help

The application installer automatically installs online help to the proper location. It is accessible from the help links within the application.

## Adding a User to the RPM Application

For LDAP authentication, complete the following steps.

1. Build/copy existing RPM user in LDAP to the new user name you desire. User in LDAP for RPM must have objectclass, retailUser, as there is a search filter on that objectclass name of retailUser.

2. Insert row to database table:

```
insert into rsm_user_role
(id, user_id, role_id, start_date_time, end_date_time)
select rsm_user_role_seq.nextval,
 'retail.user1',
 -1001,
 nvl(get_vdate,sysdate) - 365,
 null
from dual;
```



# 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 14.1.2, several additional 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"> <li>▪ Database scripts</li> <li>▪ Batch scripts</li> <li>▪ RETL scripts</li> <li>▪ Data Conversion Scripts</li> <li>▪ Forms</li> <li>▪ BI Publisher Reports</li> </ul> |
| Oracle Retail Warehouse Management System (RWMS) | <ul style="list-style-type: none"> <li>▪ Database scripts</li> <li>▪ Batch scripts</li> <li>▪ Forms</li> <li>▪ BI Publisher Reports</li> </ul>                                                            |

| Product                                                 | Supported Technology                                                                                                                    |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Oracle Retail Price Management (RPM)                    | <ul style="list-style-type: none"> <li>Database scripts (included with RMS)</li> <li>Java Application</li> <li>Batch scripts</li> </ul> |
| Oracle Retail Invoice Matching (ReIM)                   | <ul style="list-style-type: none"> <li>Database scripts (included with RMS)</li> <li>Java Application</li> <li>Batch scripts</li> </ul> |
| Oracle Retail Allocation                                | <ul style="list-style-type: none"> <li>Database scripts (included with RMS)</li> <li>Java Application</li> <li>Batch scripts</li> </ul> |
| Oracle Retail Sales Audit (ReSA)                        | <ul style="list-style-type: none"> <li>Database scripts (included with RMS)</li> <li>Java Application</li> </ul>                        |
| Oracle Retail Analytics (RA)                            | <ul style="list-style-type: none"> <li>Database scripts</li> </ul>                                                                      |
| Oracle Retail Advanced Science Engine (ORASE)           | <ul style="list-style-type: none"> <li>Database scripts</li> <li>Batch scripts</li> </ul>                                               |
| Oracle Retail Application Security Role Manager (RASRM) | <ul style="list-style-type: none"> <li>Java Application</li> </ul>                                                                      |

## 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 14.1.0 or 14.1.2 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 14.1

Many products and technologies are supported by the enhanced patching process for the first time in 14.1. In those cases all of the content in this chapter is new with 14.1.

### MMHOME changed to RETAIL\_HOME

For RMS and RWMS, which were previously supported in 14.0, there is a change when using ORPatch and related tools. Previously the MMHOME environment variable was used to refer to the RMS and RWMS installation area. Starting with 14.1, RETAIL\_HOME is now used to refer to the installation area. So where previously it was necessary to set MMHOME before executing ORPatch, you must now set RETAIL\_HOME.

---

**Note:** RMS Batch continues to use MMHOME to refer to the area where batch is installed, and requires it to be set when executing batches. The change to using RETAIL\_HOME relates only to ORPatch and related utilities.

---

### Java batch script location

For Java products with batch scripts, starting with 14.1 the location of batch scripts has been changed to \$RETAIL\_HOME/<app>-batch. Previously batch scripts were stored

within the WebLogic domain in the retail directory. Credential store files continue to be stored within the WebLogic domain.

## 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 14.1.1, 14.1.2, 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 14.1 tracks version information for all files involved with a product installation. The RETAIL\_HOME now 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.
- 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                                                                                     |
| 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                                                                                                          |

| Log File                               | Used For                                    |
|----------------------------------------|---------------------------------------------|
| 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<br>Example: orpatch apply -s \$RETAIL_HOME/stage/patch123456                                                |
| analyze        | Tells ORPatch to analyze a patch, requires the -s option<br>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 -srcname options. |
| revert         | Tells ORPatch to revert the files related to a patch, requires the -s option<br>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<br>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. |

| Argument        | Valid For Actions                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -new            | apply                                         | Forces ORPatch to not attempt to restart a failed ORPatch session                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| -expname        | exportmetadata<br>diffmetadata<br>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<br>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.<br><br>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<br>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/14.1/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/14.1/tst`

3. Set the PATH environment variable to include the orpatch/bin directory  
`export PATH=$RETAIL_HOME/orpatch/bin:$PATH`
4. Set the DISPLAY environment variable if the patch contains Forms.  
`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.  
`mkdir -p $RETAIL_HOME/stage`
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.  
`orpatch apply -s $RETAIL_HOME/stage/patch123456`
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.

### 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/orpatch/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/14.1/tst
```

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

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

4. Execute orpatch 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                         | orphatch/config/env_info.cfg<br>orphatch/config/custom_hooks.cfg<br>ORPatch inventory files | ORPatch configuration and settings                                                                                         |
| Any                         | All env_manifest.csv and<br>deleted_env_manifest.csv 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<br>javaapp_<product>/config except jar files                                 | Environment-specific product configuration files generated during installation                                             |
| 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           | orphatch/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                                                                          |

| Installed Product Component | Exported Metadata                                         | Description                                                 |
|-----------------------------|-----------------------------------------------------------|-------------------------------------------------------------|
| Java Applications           | <weblogic_home>/common/nodemanager/nodemanager.properties | Weblogic nodemanager configuration file                     |
| Java Applications           | <domain_home>/config/fmwconfig/jps-config.xml             | JPS configuration file for the Weblogic application domain. |
| 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/14.1/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/orpatch/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 "`orpatch 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 `orpatch 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/14.1/dev
```

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

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

6. Unzip the metadata zip file.

```
unzip test_env.zip
```

7. Execute `orpatch` to compare the metadata

```
orpatch 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/14.1/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  
`orphatch 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

```

ordeloy -a RPM -t JAVA
ordeloy -a REIM -t JAVA
ordeloy -a ALLOC -t JAVA
ordeloy -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.

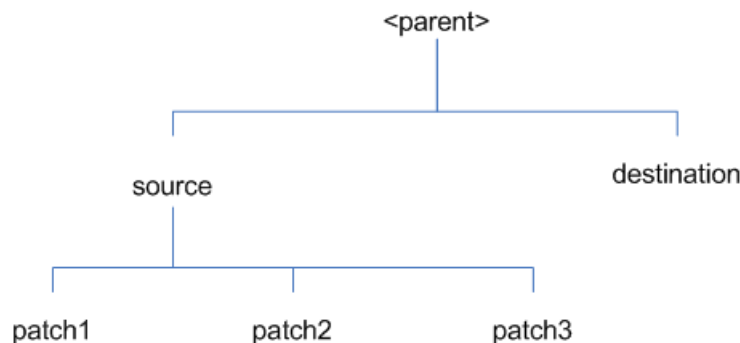
### Source and Destination Directories

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 |

| Argument | Required | Description                                                                                                                                                                                  |
|----------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -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/14.1/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              |
| RA          | DB-DM       | Compile invalid database objects in the RA DM schema            |
| RA          | DB-RABATCH  | Compile invalid database objects in the RA batch schema         |
| RA          | DB-RMSBATCH | Compile invalid database objects in the RA RMS batch schema     |
| RA          | DB-FEDM     | Compile invalid database objects in the RA front-end 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/14.1/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory  

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. 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 RA -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, <b>NOT</b> 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, <b>NOT</b> including any custom Java files.        |
| RESA        | JAVA         | Deploy the RESA Java application, including any custom Java files.                                        |
| RESA        | JAVANOCUSTOM | Deploy the RESA Java application, <b>NOT</b> 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, <b>NOT</b> 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/14.1/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<br>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              |
| \$RETAIL_HOME/orphatch/orme_wallet     | RME Database              |
| \$RETAIL_HOME/orphatch/ra_wallet       | RA 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/orphatch/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 | 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
```

```
Apapplication 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/14.1/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/14.1/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_&lt;app&gt;/base</code>     |
| 2     | Updated configuration files     | <code>\$RETAIL_HOME/javaapp_&lt;app&gt;/config</code>   |
| 3     | Oracle Retail-supplied hotfixes | <code>\$RETAIL_HOME/javaapp_&lt;app&gt;/internal</code> |
| 4     | Compiled customizations         | <code>\$RETAIL_HOME/javaapp_&lt;app&gt;/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                                                                       |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| rpm14.ear/company/ui/MyCustom.class                                         | In rpm14.ear:<br>/company/ui/MyCustom.class                                                   |
| rpm14.ear/rpm14.jar/company/bc/MyCustom2.class                              | In rpm14.ear:<br>In rpm14.jar:<br>/company/bc/MyCustom2.class                                 |
| rpm14.ear/lib/ourcustomlibs.jar                                             | In rpm14.ear<br>/lib/ourcustomlibs.jar                                                        |
| rpm14.ear/WebLaunchServlet.war/lib/<br>rpm14.jar/company/bc/MyCustom2.class | In rpm14.ear:<br>In WebLaunchServlet.war:<br>In lib/rpm14.jar:<br>/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_REIM     | Loads ReIM database objects into the RMS schema                                                  |
| 4     | DBSQL_RAF      | Loads Retail Application Framework 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                                |
| 13    | DBSQL_RWMS     | Loads database objects into the primary RWMS schema                                              |

| Order | Action Name      | Description                                                                 |
|-------|------------------|-----------------------------------------------------------------------------|
| 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 RASRM Java application                                          |
| 22    | DBSQL_RARMSBATCH | Loads database objects into the RMS Batch schema for RA                     |
| 23    | DBSQL_RADM       | Loads database objects into the RA Data Mart schema                         |
| 24    | DBSQL_RAFEDM     | Loads database objects into the RA Front-end schema                         |
| 25    | DBSQL_RABATCH    | Loads database objects into the RA Batch schema                             |
| 26    | DBSQL_RASECORE   | Loads core database objects into the ORASE schema                           |
| 27    | DBSQL_RASEASO    | Loads ASO database objects into the ORASE schema                            |
| 28    | DBSQL_RASECDT    | Loads CDT database objects into the ORASE schema                            |
| 29    | DBSQL_RASECIS    | Loads CIS database objects into the ORASE schema                            |
| 30    | DBSQL_RASEDT     | Loads DT database objects into the ORASE schema                             |
| 31    | DBSQL_RASEMBA    | Loads MBA database objects into the ORASE schema                            |
| 32    | RASECOREBATCH    | Copies ORASE core batch scripts and libraries                               |
| 33    | RASEASOBATCH     | Copies ORASE ASO batch scripts and libraries                                |
| 34    | RASECDTBATCH     | Copies ORASE CDT batch scripts and libraries                                |
| 35    | RASECISBATCH     | Copies ORASE CIS batch scripts and libraries                                |
| 36    | RASEDTBATCH      | Copies ORASE DT batch scripts and libraries                                 |
| 37    | 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. |

| Restart Phase Number | Phase Name  | Description                                                                                                                                                                                                          |
|----------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 ORPatchactivities 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                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------|
| -f <file>    | Adds <file> to the list of files that will be registered. Can be specified more than once.                         |
| -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 in the dbmanifest table                                        |
| -unregister  | Files specified with -f or -bulk 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/14.1/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/oretail14.1

---

2. Locate the patch\_info.cfg within the product media. The directory it resides in will be used for later steps.

```
find /media/oretail14.1/rms/installer -name patch_info.cfg
```

Output Example:

```
/media/oretail14.1/rms/installer/mom14/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/oretail14.1/rms/installer/mom14/patch_info.cfg
```

Output Example:

```
PATCH_NAME=MOM_14_1_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/oretail14.1/rms/installer/mom14/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/oretail14.1/rms/installer/mom14 -name MOM_14_1_0_0 -inplace
```

---

**Note:** The -inplace argument is critical to ensure that the  
patching replaces files in the mom14 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/oretail14.1/rms/installer/mom14/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

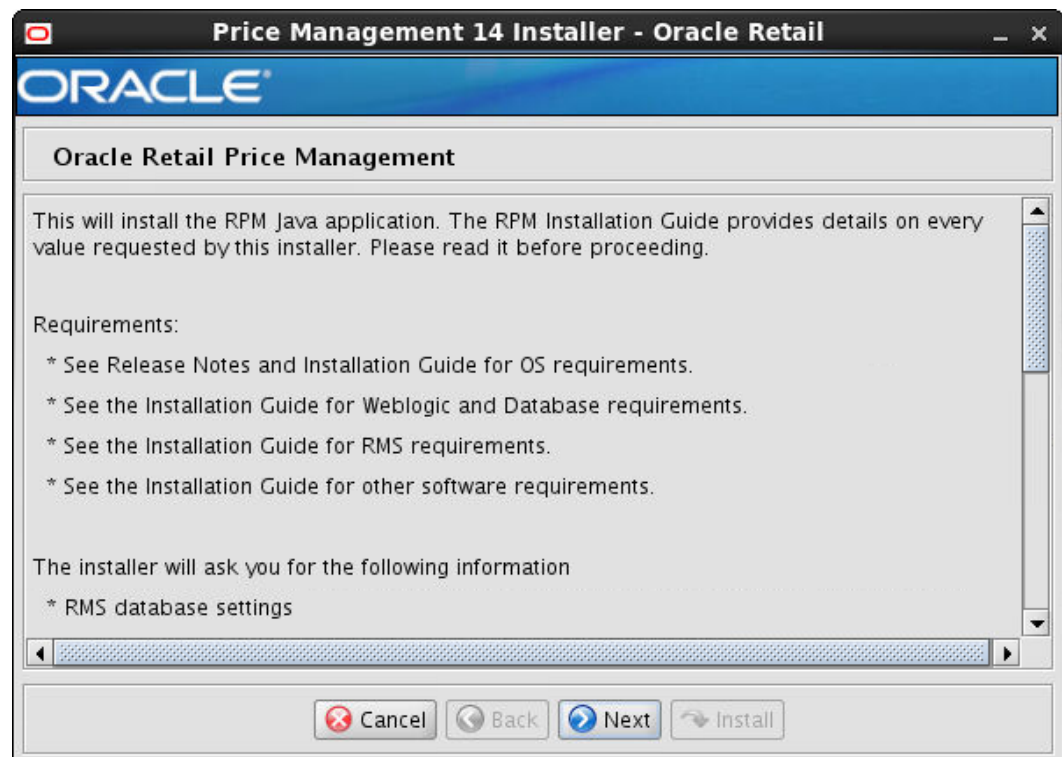
Oracle Support or for external analysis. For more information, see the ORPatch 'Exporting Environment Metadata' section.

---

## Appendix: RPM Application Installer Screens

You need the following details about your environment for the installer to successfully deploy the RPM application. Depending on the options you select, you may not see some screens or fields.

### Screen: Installation Introduction Screen



**Screen: RPM Application RETAIL HOME**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**RPM Application RETAIL\_HOME**

Please enter the directory where the RPM application files and batch scripts will be installed. Please keep track of this directory, it should remain in place after installation and will be used to apply future patches.

RPM Application RETAIL\_HOME

|                          |                                                                                                                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM Application RETAIL HOME                                                                                                                                                                              |
| <b>Field Description</b> | Retail Home is used to keep Orpatch related files, batches, etc. by default. Please keep track of this directory, it should remain in place after installation and will be used to apply future patches. |
| <b>Examples</b>          | /path/to/rpm_retail_home                                                                                                                                                                                 |

**Screen: Host Details**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Host Details**

Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname

|                          |                                                                                                                           |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Hostname                                                                                                                  |
| <b>Field Description</b> | Provide the hostname where the Retail Home, batches will be installed. This shall match your Application server hostname. |
| <b>Examples</b>          | apphostname                                                                                                               |

**Screen: Security Details**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Security Details**

Provide security details for the RPM application

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.

Enable SSL for RPM?

☒ Yes

☐ No

Cancel Back Next Install

|                          |                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Enable SSL for RPM?                                                                                                                                                                                                                                                                                                                                           |
| <b>Field Description</b> | <p>Choosing Yes will deploy RPM using SSL and configure RPM to use SSL. In this case, SSL must be configured and the ports must be enabled for the AdminServer and RPM managed servers.</p> <p>Choosing No will deploy and configure RPM without SSL. In this case the non-SSL ports must be enabled for the AdminServer and for the RPM managed servers.</p> |



**Screen: JDBC Security Details**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**JDBC Security Details**

Note: Enabling Secure JDBC requires that security certificates have been configured and installed for this WebLogic domain.

Enable Secure JDBC connection

☒ Yes  
☐ No

|                          |                                                                                                                                                                              |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Enable Secure JDBC connection                                                                                                                                                |
| <b>Field Description</b> | Choose Yes to create secured data sources in WebLogic, otherwise choose No. A secure data base connection must already be set up if you want to create a secure data source. |

**Screen: Configure Automatic Jar Signing**

|                          |                                                                                                                                                                                                                                                                             |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Configure automatic jar signing                                                                                                                                                                                                                                             |
| <b>Field Description</b> | Choosing Yes will enable the auto signing of the rpm client config jar with the keystore that will be asked in the next screen.<br>Choosing No will skip the Auto Jar Signing Details. In this case, rpm client config jar shall be signed manually after the installation. |

**Screen: Auto Jar Signing Details**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Auto Jar Signing Details**

Provide the details for the automatic signing of Jars

JAVA Keystore Location: /u00/webadmin/product/rpm\_orp\_keyst

Passphrase for keystore: .....

Passphrase for private key: .....

JAVA Keystore Alias: orpatchjarkey

Buttons: Cancel, Back, Next, Install

---

**Note:** This screen will appear only if you select Configure automatic Jar signing in the above screen.

---

|                          |                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Java Keystore Location                                                                                                                |
| <b>Field Description</b> | This is the path of the keystore which contains the ssl identity certificates of the host as obtained from the certificate authority. |
| <b>Example</b>           | /u00/webadmin/product/rpm_orp_keystore/orpkeystore.jks                                                                                |

|                          |                                    |
|--------------------------|------------------------------------|
| <b>Field Title</b>       | Passphrase for keystore            |
| <b>Field Description</b> | The password for the keystore used |

|                          |                                       |
|--------------------------|---------------------------------------|
| <b>Field Title</b>       | Passphrase for private key            |
| <b>Field Description</b> | The password for the private key used |

|                          |                                                             |
|--------------------------|-------------------------------------------------------------|
| <b>Field Title</b>       | JAVA Keystore Alias                                         |
| <b>Field Description</b> | Alias of the identity certificate inside the keystore file. |
| <b>Example</b>           | Orpatchjarkey                                               |

## Screen: Data Source Details

**Price Management 14 Installer - Oracle Retail**

**ORACLE®**

**Data Source Details**

Provide the details for the RPM data source

RMS 14 JDBC URL

RPM/RMS 14 schema user

RPM/RMS 14 schema password

Enter the RMS schema owner. This is usually the same as the RMS schema entered above

RMS 14 schema owner

Note: entering an alias for this user will enhance security for this application. If left blank it will default to the username.

RPM/RMS 14 schema user alias

|                          |                                                                                                                                                                                                                                                 |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | RMS 14 JDBC URL                                                                                                                                                                                                                                 |
| <b>Field Description</b> | URL used by the RPM application to access the RMS database schema. See <a href="#">Appendix: URL Reference</a> for expected syntax.<br><b>Note:</b> The RPM database tables are a part of the RMS schema.                                       |
| <b>Examples</b>          | For Non Secure JDBC Connection: jdbc:oracle:thin:@hostname:1521/dbname<br>For Secure JDBC Connection:<br>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcps)(HOST=dbhostname)(PORT=2484)))(CONNECT_DATA=(SERVICE_NAME=mydb))) |

|                          |                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM/RMS 14 schema user                                                                                                                  |
| <b>Field Description</b> | RMS database user for accessing the RPM tables. This should match what was given in the RMS schema field of the RMS database installer. |
| <b>Example</b>           | rms01app                                                                                                                                |

|                          |                                                                            |
|--------------------------|----------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM/RMS 14 schema password                                                 |
| <b>Field Description</b> | Password for the RMS database user entered above to access the RPM tables. |

|                          |                                                                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | RMS 14 schema owner                                                                                   |
| <b>Field Description</b> | Database user which owns the RMS and RPM tables. This is usually the same as the RMS 14 schema above. |
| <b>Example</b>           | rms01                                                                                                 |

|                          |                                                                                                                                                                                                                     |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM/RMS 14 schema alias                                                                                                                                                                                             |
| <b>Field Description</b> | The alias to store the schema credentials.                                                                                                                                                                          |
| <b>Example</b>           | db-alias                                                                                                                                                                                                            |
| <b>Notes</b>             | This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application. |

**Screen: Secure Data Source Details**

**Price Management 14 Installer - Oracle Retail**

**ORACLE®**

**Secure Data Source Details**

Provide the details for the RPM secure data source

Identity Keystore: /path/sample.keystore

Identity Keystore Type: JKS

Identity Keystore Passphrase: .....

Identity truststore: /path/test.keystore

Identity truststore Type: JKS

Identity truststore Passphrase: .....

Cancel Back Next Install

**Note:** This screen will appear only if you select Secure JDBC in the above screens.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Identity Keystore                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Field Description</b> | Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This screen lets you provide the keystore to be used for datasource connection. These settings help you to manage the security of message transmissions. For further information, please refer to the <i>Oracle Retail Merchandising Operations Management Security Guide</i> .<br>Location or path where identity keystore file is stored. |
| <b>Example</b>           | /path/sample.keystore                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                          |                                |
|--------------------------|--------------------------------|
| <b>Field Title</b>       | Identity Keystore Type         |
| <b>Field Description</b> | The type of the keystore used. |
| <b>Example</b>           | JKS                            |

|                          |                                                                 |
|--------------------------|-----------------------------------------------------------------|
| <b>Field Title</b>       | Identity Keystore PassPhrase                                    |
| <b>Field Description</b> | Please provide password to access the keystore mentioned above. |

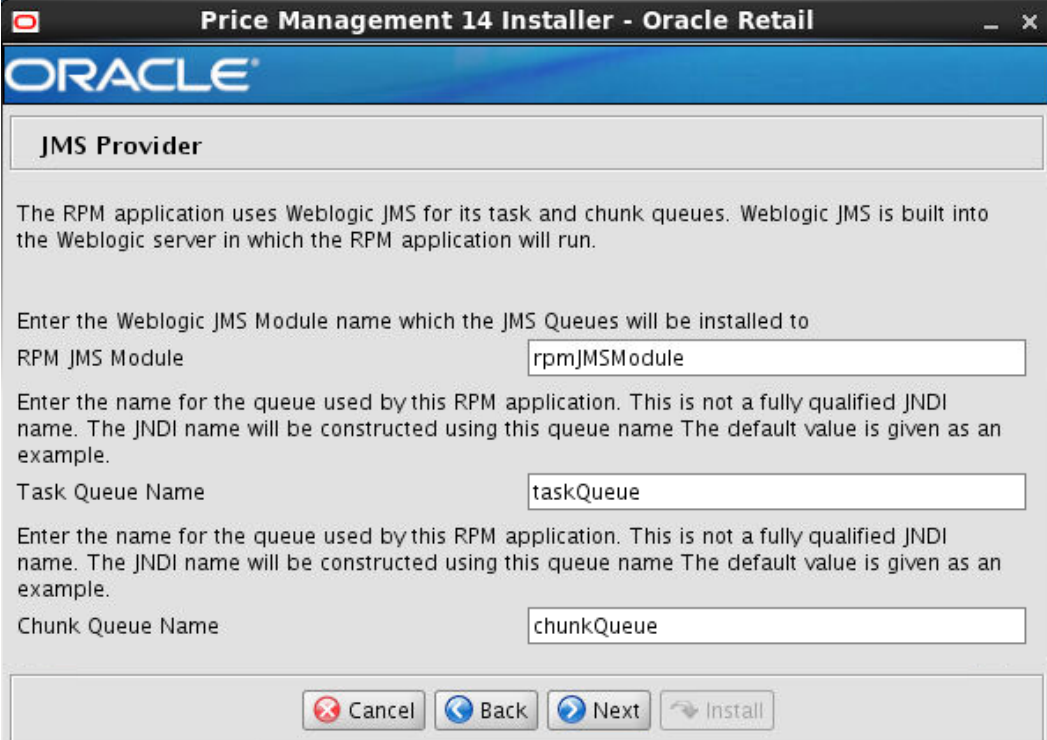
|                          |                                                                                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Identity TrustStore                                                                                                                               |
| <b>Field Description</b> | This is the path of the keystore which contains the ssl root and optionally intermediate certificates as obtained from the certificate authority. |
| <b>Example</b>           | /path/test.keystore                                                                                                                               |

|                          |                                 |
|--------------------------|---------------------------------|
| <b>Field Title</b>       | Identity TrustStore Type        |
| <b>Field Description</b> | The type of the truststore used |
| <b>Example</b>           | JKS                             |

|                          |                                                                   |
|--------------------------|-------------------------------------------------------------------|
| <b>Field Title</b>       | Identity TrustStore PassPhrase                                    |
| <b>Field Description</b> | Please provide password to access the truststore mentioned above. |



## Screen: JMS Provider



**JMS Provider**

The RPM application uses Weblogic JMS for its task and chunk queues. Weblogic JMS is built into the Weblogic server in which the RPM application will run.

Enter the Weblogic JMS Module name which the JMS Queues will be installed to





RPM JMS Module

Enter the name for the queue used by this RPM application. This is not a fully qualified JNDI name. The JNDI name will be constructed using this queue name The default value is given as an example.

Task Queue Name

Enter the name for the queue used by this RPM application. This is not a fully qualified JNDI name. The JNDI name will be constructed using this queue name The default value is given as an example.

Chunk Queue Name

 Cancel  Back  Next  Install

|                          |                                                                         |
|--------------------------|-------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM JMS Module                                                          |
| <b>Field Description</b> | The WebLogic JMS Module name to which the JMS Queues will be installed. |
| <b>Example</b>           | rpmJMSModule                                                            |

|                          |                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Task Queue Name                                                                                                                                                                                                                                                                                                                       |
| <b>Field Description</b> | Name by which the task queue will be identified. If this is a new RPM environment, choose a queue name that is not already in use in the JMS server. If you have already created the queue in the JMS server as part of the Clustering Preinstallation steps, you must provide the same name in this field (without the jms/ prefix). |
| <b>Example</b>           | taskQueue                                                                                                                                                                                                                                                                                                                             |

|                          |                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Chunk Queue Name                                                                                                                                                                                                                                                                                                                      |
| <b>Field Description</b> | Name by which the task queue will be identified. If this is a new RPM environment, choose a queue name that is not already in use in the JMS server. If you have already created the queue in the JMS server as part of the Clustering Preinstallation steps, you must provide the same name in this field (without the jms/ prefix). |
| <b>Example</b>           | chunkQueue                                                                                                                                                                                                                                                                                                                            |

## Screen: LDAP directory server details

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**LDAP directory server details**

Note: If the ldap server is configured to use SSL, use ldaps as the protocol. Otherwise use ldap.

LDAP server URL

Enter the search user DN. RPM will authenticate to the LDAP directory as this entry.

Search User DN

Search User Password

Note: entering an alias for this user will enhance security for this application. If left blank it will default to the username.

Search User Alias

|                          |                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP server URL                                                                     |
| <b>Field Description</b> | URL for your LDAP directory server.                                                 |
| <b>Example</b>           | For Non Secure LDAP: ldap://hostname:3060<br>For Secure LDAP: ldaps:// hostname:389 |

|                          |                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Search User DN                                                                      |
| <b>Field Description</b> | Distinguished name of the user that RPM uses to authenticate to the LDAP directory. |
| <b>Example</b>           | cn=RPM.ADMIN,cn=Users,dc=us,dc=oracle,dc=com                                        |

|                          |                                  |
|--------------------------|----------------------------------|
| <b>Field Title</b>       | Search User Password             |
| <b>Field Description</b> | Password for the search user DN. |

|                          |                                                                                                                                                                                                                     |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Search User Alias                                                                                                                                                                                                   |
| <b>Field Description</b> | The alias for the search user DN.                                                                                                                                                                                   |
| <b>Example</b>           | ldap-alias                                                                                                                                                                                                          |
| <b>Notes</b>             | This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application. |

## Screen: LDAP directory server searches

**Price Management 14 Installer - Oracle Retail**

**LDAP directory server searches**

Enter the search base DN. This is a directory entry under which RPM will search for groups

LDAP search base DN

Enter the LDAP search filter for RPM to use when performing LDAP searches

LDAP search filter

attribute for usernames

The LDAP user group that contains all RPM application users.

LDAP Secure Users Group

An LDAP user that is required for propagation of security privileges. It must belong to the user group that contains all RPM application users.

LDAP Secure Resource User

The main filtering attribute to locate users on the LDAP.

|                          |                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP search base DN                                                                     |
| <b>Field Description</b> | Distinguished name of the LDAP directory entry under which RPM should search for users. |
| <b>Example</b>           | dc=us,dc=oracle,dc=com                                                                  |

|                          |                                                                                                                                                                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP search filter                                                                                                                                                                                                                                  |
| <b>Field Description</b> | LDAP filter that determines which entries are returned to RPM when it conducts a directory search under the search base DN.<br>See the <i>Oracle Retail Price Management Operations Guide</i> for additional information on configuring this field. |
| <b>Example</b>           | (&(objectclass=retailUser) %v)                                                                                                                                                                                                                      |

|                          |                                                             |
|--------------------------|-------------------------------------------------------------|
| <b>Field Title</b>       | attribute for usernames                                     |
| <b>Field Description</b> | LDAP attribute where RPM should look for a user's username. |
| <b>Example</b>           | uid                                                         |

|                          |                                                                     |
|--------------------------|---------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP Secure Users Group                                             |
| <b>Field Description</b> | The LDAP group that contains all the Users that can login into RPM. |
| <b>Example</b>           | rpm_secure_users                                                    |

|                          |                                                                                                                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP Secure Resource User                                                                                                                                                                        |
| <b>Field Description</b> | Distinguished name of the user that RPM uses to authenticate to the LDAP directory and propagation of Security privileges. It must be in the user group that contains all RPM application users. |
| <b>Example</b>           | rpm.admin                                                                                                                                                                                        |

|                          |                                                                     |
|--------------------------|---------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP Search Filter Attribute                                        |
| <b>Field Description</b> | LDAP filtering attribute used by RPM to filter all the valid users. |
| <b>Example</b>           | Objectclass                                                         |

|                          |                                                                                    |
|--------------------------|------------------------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP Search Filter Value                                                           |
| <b>Field Description</b> | The criteria value of the LDAP filtering attribute that a valid user should match. |
| <b>Example</b>           | retailUser                                                                         |

|                          |                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | LDAP Identity Story Factory Class                                                                        |
| <b>Field Description</b> | The factory class that matches the LDAP Provider used to store identity information for the application. |
| <b>Example</b>           | Oracle Internet Directory                                                                                |

## Screen: RPM UI Client

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**RPM UI Client**

Please enter the web context root for the RPM client files.

Client Context Root

Use Oracle Single Sign-On for user identification and authentication?

☒ Yes. OSSO will provide the user name.

☐ No. The user will provide this information.

Oracle Single Sign-On must be installed separately and the HTTP Server used to download the RPM client must be registered with the OSSO server before you can use it.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Client Context Root                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Field Description</b> | <p>The Client Context Root determines how the RPM client will be accessed from users' web browsers. The RPM client URL has the following format:</p> <p><code>http://&lt;hostname&gt;:&lt;port&gt;/&lt;rpm_client_ctx_root&gt;/launch?template=rpm_jnlp_template.vm</code></p> <p>Example, with RPM Client Context Root value of rpm-client:</p> <p><a href="http://apphostname:17011/rpm-client/launch?template=rpm_jnlp_template.vm">http://apphostname:17011/rpm-client/launch?template=rpm_jnlp_template.vm</a></p> |
| <b>Example</b>           | rpm-client                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|                          |                                                                                                                                                                                                                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Use Oracle Single Sign-On for user identification and authentication?                                                                                                                                                                                                                |
| <b>Field Description</b> | <p>This version of RPM has the option to use Oracle Access Manager (Webgate Agent) technology to authenticate users.</p> <p>If OAM is being used in your environment, choose Yes.</p> <p>The No option configures RPM to use its own LDAP directory settings for authentication.</p> |
| <b>Example</b>           | No                                                                                                                                                                                                                                                                                   |

## Screen Single Sign-On Details

**Note:** This screen will only be displayed if the SSO option was selected in the previous step.

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Oracle Single Sign-On Details**

Please enter the Oracle Single Sign-On web tier Details.

OSSO web tier Server: Appserver1.us.oracle.com

OSSO web tier port: 18888

SSO token generation key Alias: SSO-TOKEN-KEY-ALIAS

Select SSO token key generation type, If you select Yes The token generation key will be regenerated on server start-up enhancing security (Recommended) and if you select no the token generation key must be inserted and managed manually in the credential store (Not recommended).

SSO token key generation type: ☒ Yes. Generated on server startup ☐ No. Managed manually

Buttons: Cancel, Back, Next, Install

|                          |                                                                      |
|--------------------------|----------------------------------------------------------------------|
| <b>Field Title</b>       | OSSO Web Tier Server                                                 |
| <b>Field Description</b> | This should have the host name on which the web tier is deployed on. |
| <b>Example</b>           | Appserver1.us.oracle.com                                             |

|                          |                                                                   |
|--------------------------|-------------------------------------------------------------------|
| <b>Field Title</b>       | OSS Web Tier port                                                 |
| <b>Field Description</b> | The HTTP port of the webtier installation must be mentioned here. |
| <b>Example</b>           | 18888                                                             |



|                          |                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------|
| <b>Field Title</b>       | SSO Token generation key alias                                                          |
| <b>Field Description</b> | SSO uses this to store its tokens that are used to verify authenticity of the SSO call. |
| <b>Example</b>           | SSO-TOKEN-KEY-ALIAS                                                                     |

|                          |                                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | SSO Token generation key type                                                                             |
| <b>Field Description</b> | If you want to have the token generation by manual or by installer. If manual select No. else select Yes. |
| <b>Example</b>           | Yes                                                                                                       |

## Screen Web Tier Details

Price Management 14 Installer - Oracle Retail

ORACLE

Oracle Web tier details

Do you wish to enter web tier details ?

☒ Yes. I want to enter web tier detail

☐ No. I do not need webtier

Cancel Back Next Install

---

**Note:** This screen will appear only if SSO is disabled in screen "RPM UI Client"

---

|                   |                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------|
| Field Title       | Do you wish to enter Web Tier details?                                                                         |
| Field Description | Yes – If user wishes to use webtier without SSO<br>No – If user wishes to continue without SSO without webtier |

## Screen Enter Web Tier Details

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Enter Webtier Details**

If you choose, http we understand that you have configured secured part manually.  
Please choose protocol for webtier.

☒ https  
☐ http

Web tier Server

Web tier port

Cancel Back Next Install

**Note:** This screen will appear only if “yes” is selected in above screen

|                          |                                     |
|--------------------------|-------------------------------------|
| <b>Field Title</b>       | Please choose protocol for web-tier |
| <b>Field Description</b> | Protocol for web-tier               |
| <b>Examples</b>          | http/https                          |

|                          |                              |
|--------------------------|------------------------------|
| <b>Field Title</b>       | Web Tier Server              |
| <b>Field Description</b> | Server name hosting web tier |
| <b>Examples</b>          | apphost                      |

|                          |                       |
|--------------------------|-----------------------|
| <b>Field Title</b>       | Web Tier Port         |
| <b>Field Description</b> | Port running web-tier |
| <b>Examples</b>          | 1521                  |

**Screen: Installation Type**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Installation Type**

The RPM application can be installed on two types of servers Standalone server or Cluster servers. The default Installation is Standalone server, alternatively you can choose cluster installation

Which Installation method will you use?

☒ Standalone server

☐ Cluster servers

Cancel Back Next Install

|                          |                                                                                                        |
|--------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Installation type                                                                                      |
| <b>Field Description</b> | The default installation type is standalone server; alternatively you can choose cluster installation. |

**Screen: Application Deployment Details**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Application Deployment Details**

The default values shown below are examples

RPM app deployment name

Enter the RPM weblogic managed server or cluster.

RPM server/cluster

|                          |                                                                             |
|--------------------------|-----------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM 14 app deployment name                                                  |
| <b>Field Description</b> | Name by which this RPM application is identified in the application server. |
| <b>Example</b>           | rpm14                                                                       |

|                          |                                                                                                                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | RPM 14 server/cluster                                                                                                                                                                                                                                                                   |
| <b>Field Description</b> | Name of the server/cluster that was created for this RPM application.<br>The installer deploys the RPM application to all instances that are members of this server/cluster. For this reason, you should not use default_group. A new group dedicated to RPM should be created instead. |
| <b>Example</b>           | Rpm_server1                                                                                                                                                                                                                                                                             |

## Screen: WebLogic Administrative Details

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Weblogic Administrative Details**

Enter the administrative user and password for the Weblogic Server to which the application will be deployed.

Note:if SSL is enabled, this value MUST match the DNS name used in the SSL certificate.

Weblogic hostname: apphostname

Weblogic admin user: weblogic

Weblogic admin password: .....

Weblogic admin alias: wls-alias

Buttons: Cancel, Back, Next, Install

|                          |                                         |
|--------------------------|-----------------------------------------|
| <b>Field Title</b>       | Hostname                                |
| <b>Field Description</b> | The Hostname of the application server. |
| <b>Example</b>           | apphostname                             |

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | WebLogic admin user                                                                                  |
| <b>Field Description</b> | Username of the admin user for the WebLogic instance to which the RPM application is being deployed. |
| <b>Example</b>           | weblogic                                                                                             |

|                          |                                                                                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | WebLogic admin password                                                                                                                                   |
| <b>Field Description</b> | Password for the WebLogic admin user. You chose this password when you created the WebLogic instance or when you started the instance for the first time. |

|                          |                                                                                                                                                                                                                     |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | WebLogic admin alias                                                                                                                                                                                                |
| <b>Field Description</b> | An alias for the WebLogic admin user that is used for ORACLE java wallet.                                                                                                                                           |
| <b>Example</b>           | wls-alias                                                                                                                                                                                                           |
| <b>Notes</b>             | This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application. |



**Screen: Batch User Credentials**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Batch User Credentials**

Provide the credentials for the Batch User

Note: this must be a valid rsm/rpm user.

Batch user: RETAIL.USER

Batch User password: .....

Cancel Back Next Install

|                          |                                                                                                                  |
|--------------------------|------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Batch User                                                                                                       |
| <b>Field Description</b> | The RPM user name of the person running RPM batch. It must be a valid RPM user that will be coming through LDAP. |
| <b>Example</b>           | RETAIL.USER                                                                                                      |

|                          |                                 |
|--------------------------|---------------------------------|
| <b>Field Title</b>       | Batch User Password             |
| <b>Field Description</b> | The password of the batch user. |

**Screen: Choose Apps to Integrate with RPM**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Choose Apps to Integrate with RPM**

Choose which applications you would like to integrate with RPM.

Configure RIB for RPM? ☐

Cancel Back Next Install

|                          |                                                       |
|--------------------------|-------------------------------------------------------|
| <b>Field Title</b>       | Configure RIB for RPM?                                |
| <b>Field Description</b> | Select this option if you will be using RIB with RPM. |

## Screen: RIBforRPM Details

**Price Management 14 Installer - Oracle Retail**

**ORACLE®**

**RIBforRPM Details**

If RPM will be integrated with RIB, then provide the details (Optional).

The app-level partition (mapname) for the credentials will be set to rpm.

rib-rpm Weblogic User:

rib-rpm Weblogic Password:

Note: entering an alias for this user will enhance security for this application. If left blank it will default to username.

rib-rpm Weblogic Alias:

Note: If rib-rpm uses SSL, use t3s as the protocol. Otherwise use t3.

rib-rpm Provider Url:

Buttons: Cancel, Back, Next, Install

**Note:** This screen will only be displayed if the check box was checked in the screen prior to this one.

|                          |                                                         |
|--------------------------|---------------------------------------------------------|
| <b>Field Title</b>       | rib-rpm WebLogic User                                   |
| <b>Field Description</b> | The username of the server where rib-rpm is configured. |
| <b>Example</b>           | weblogic                                                |

|                          |                                                      |
|--------------------------|------------------------------------------------------|
| <b>Field Title</b>       | rib-rpm WebLogic password                            |
| <b>Field Description</b> | Password for the server where rib-rpm is configured. |

|                          |                                                                                                                                                                                                                     |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | rib-rpm WebLogic Alias                                                                                                                                                                                              |
| <b>Field Description</b> | The alias for the rib-rpm WebLogic credentials.                                                                                                                                                                     |
| <b>Example</b>           | rib-wls-alias                                                                                                                                                                                                       |
| <b>Notes</b>             | This alias must be unique. Do not use the same value for any other alias fields in the installer. If the same alias is used, entries in the wallet can override each other and cause problems with the application. |

|                          |                                      |
|--------------------------|--------------------------------------|
| <b>Field Title</b>       | rib-rpm Provider URL                 |
| <b>Field Description</b> | RPM provider URL for rib-rpm server. |
| <b>Examples</b>          | t3s://myhost:8005/rib-rpm            |

**Screen: Turn off the application servers's non-SSL port**

**Price Management 14 Installer - Oracle Retail**

**ORACLE**

**Turn off the application server's non-SSL port**

If turned off, all clients connecting to the application server must use a secured connection.

A value of "Yes" indicates that the application server's non-SSL port will be inactive. A value of "No" indicates that the applications server's non-SSL port will still be active.

Disable non-SSL port? ☒ Yes ☐ No

Cancel Back Next Install

---

**Note:** This screen appears only if you have enabled SSL for RPM. Ignore this step in case you have not enabled SSL for RPM.

---

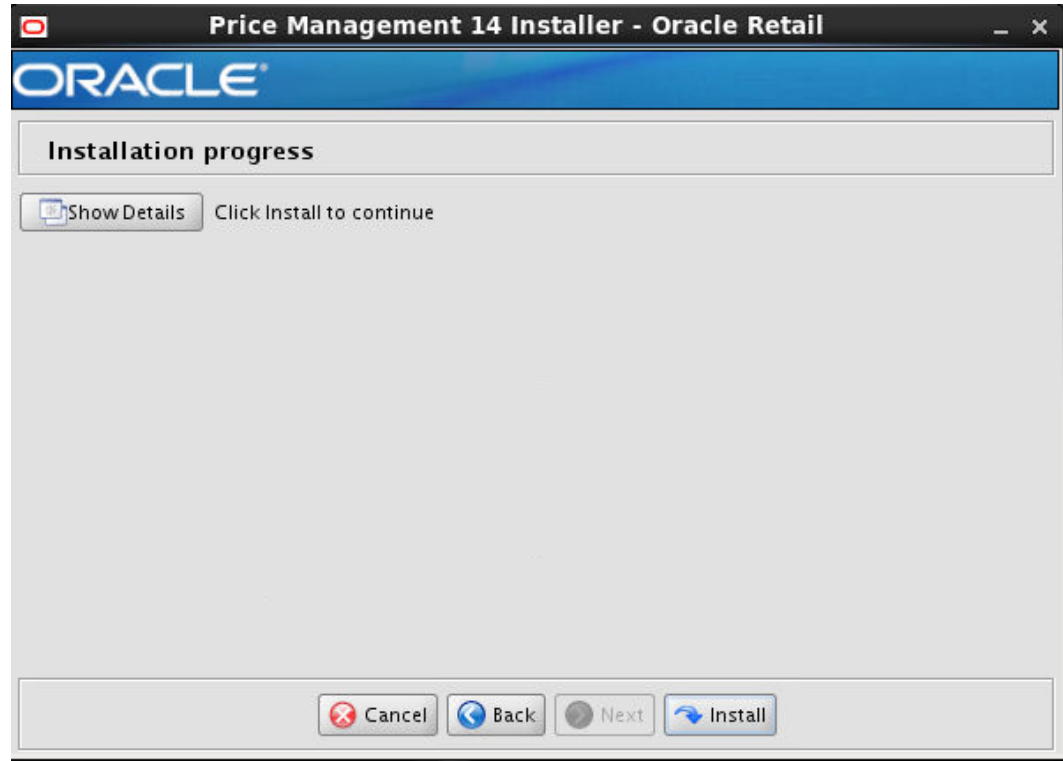
|                          |                                                                                                                                            |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Field Title</b>       | Disable non-SSL port?                                                                                                                      |
| <b>Field Description</b> | Choosing Yes disables the non SSL port on the managed server.<br>Choosing no will the leave the non SSL port of the managed server active. |

**Screen: Installation Summary**

The screenshot shows the 'Installation Summary' window of the 'Price Management 14 Installer - Oracle Retail'. The window has a blue header with the Oracle logo. Below the header, the title 'Installation Summary' is displayed. The main area is titled 'Summary of Installation' and contains a list of configuration parameters and their values. At the bottom, there are four buttons: 'Cancel', 'Back', 'Next', and 'Install'.

| Summary of Installation      |                                       |
|------------------------------|---------------------------------------|
| Hostname                     | apphostname                           |
| RPM Application RETAIL_HOME  | k/pkg_mock/rpm141/RC1/retail_home     |
| Enable SSL for RPM           | true                                  |
| Enable SecureJDBC for RPM    | true                                  |
| Enable Automatic Jar Signing | true                                  |
| JAVA Keystore Location       | duct/rpm_orp_keystore/orpkeystore.jks |
| JAVA Keystore Alias          | Orpatchjarkey                         |
| Data Source URL              | DATA = (SERVICE_NAME = polsp02app)))  |
| Data Source Username         | rms01app                              |

Buttons:

**Screen: Installation Progress**





---

## Appendix: Common Installation Errors

---

This section provides some common errors encountered during installation of RPM.

### Keystore errors when signing rpm\_client\_config.jar

#### Error message

keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect

This message may be encountered when you use the **keytool** utility to create an alias for signing the rpm\_client\_config.jar file. This usually happens when the alias for which you are generating a key already exists in the keystore file.

#### Solution

Delete or rename the ~/.keystore file and run the keytool command again. This creates a fresh keystore file.

### Unreadable buttons in the Installer

If you are unable to read the text within the installer buttons, it could mean that your JAVA\_HOME is pointed to an older version of the JDK that is supported by the installer. "Set JAVA\_HOME with the appropriate JDK (the same jdk that has been used by WebLogic Server)."

### Left menu buttons missing in RPM Client

#### Symptom

You can log into the RPM application but the left-side menus do not show up on the screen.

#### Solution

The RSM (Security Manager) schema has not been loaded with RPM security data. There is a set of RPM data scripts that is shipped with RMS 14 (See Chapter 2, "[RAC and Clustering](#)"). Run these scripts in the RSM schema and try logging into RPM again.

## 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.

## Failed RPM Login

### Symptom

You will receive errors when the RPM client tries to connect to the ldap server to authenticate the user.

### Solution

Add the following tag to the **server start parameters** of the rpm managed server.

```
-Djava.security.auth.login.config=<WEBLOGIC_DOMAIN_HOME>/retail/<RPM APP
Name>/config/rpm_jaas.config
```

Validate the location of rpm\_jaas.config. Make sure weblogic.policy has the appropriate values, as specified in the [Start the Managed Servers](#) section.

## GUI screens fail to open when running Installer

### Symptom

When running the installer in GUI mode, the screens fail to open and the installer ends, returning to the console without an error message. The ant.install.log file contains this error:

```
Fatal exception: Width (0) and height (0) cannot be <= 0
java.lang.IllegalArgumentException: Width (0) and height (0) cannot be <= 0
```

### Solution

This error is encountered when Antinstaller is used in GUI mode with certain X Servers. To work around this issue, copy ant.install.properties.sample to ant.install.properties and rerun the installer.

---

## Appendix: URL Reference

The application installer for the RPM product asks for several different URLs. These include the following.

### 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

---

### JNDI Provider URL for an Application

Used by the application client to access the application running in the server. This is also used by other applications for server-to-server calls.

Syntax: t3://<host>:<port>/<app>

- <host>: hostname of the WebLogic environment
- <port>: Port of the managed server to which rpm has been deployed. This can be found in the <WEBLOGIC\_DOMAIN\_HOME>/config/config.xml file.
- <app>: Deployment name for the application.

---

**Example:** t3://myhost:17011/rpm14

---

**Note:** The JNDI provider URL can have a different format depending on your cluster topology. Consult the WebLogic documentation.

---



---

## 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/rms14/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/rms14/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/11.2.0.1/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/rms14/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/rms14/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\_ALIAS should point to the Java wallet entry:

- export RETL\_WALLET\_ALIAS="retl\_java\_rms01user"
  - The ORACLE\_WALLET\_ALIAS should point to the Oracle network wallet entry:
    - export ORACLE\_WALLET\_ALIAS="dvols29\_rms01user"
  - The SQLPLUS\_LOGON should use the ORACLE\_WALLET\_ALIAS:
    - export SQLPLUS\_LOGON="/@\${ORACLE\_WALLET\_ALIAS}"
- 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/product/10.3.6/WLS/user\_projects/domains/14\_mck\_soa\_domain/retail/reim14/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/product/10.3.6/WLS/user\_projects/domains/REIMDomain/retail/reim14/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/reim14/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/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/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/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/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/product/10.3.x/WLS/user_projects/domains/REIMDomain/retail/reim14/config
```

Retail Public Security API Utility

```
=====
```

Below are the credentials found in the wallet at the  
location: /u00/webadmin/product/10.3.x/WLS/user\_projects/domains/REIMDomain/retail/reim14/config

```
=====
```

```
Application level key partition name:reim14
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:rms14mock
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/mock14_testing/rtil/rtil/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=rms14mock
datasource.credential.alias=RMS-ALIAS
=====
ossa related Configuration
#
These settings are for ossa configuration to store credentials.
=====

csm.wallet.path=/u00/webadmin/product/10.3.x/WLS/user_projects/domains/REIMDomain/
retail/reim14/config
csm.wallet.partition.name=reim14
```

## How does the Wallet Relate to Java Batch Program use?

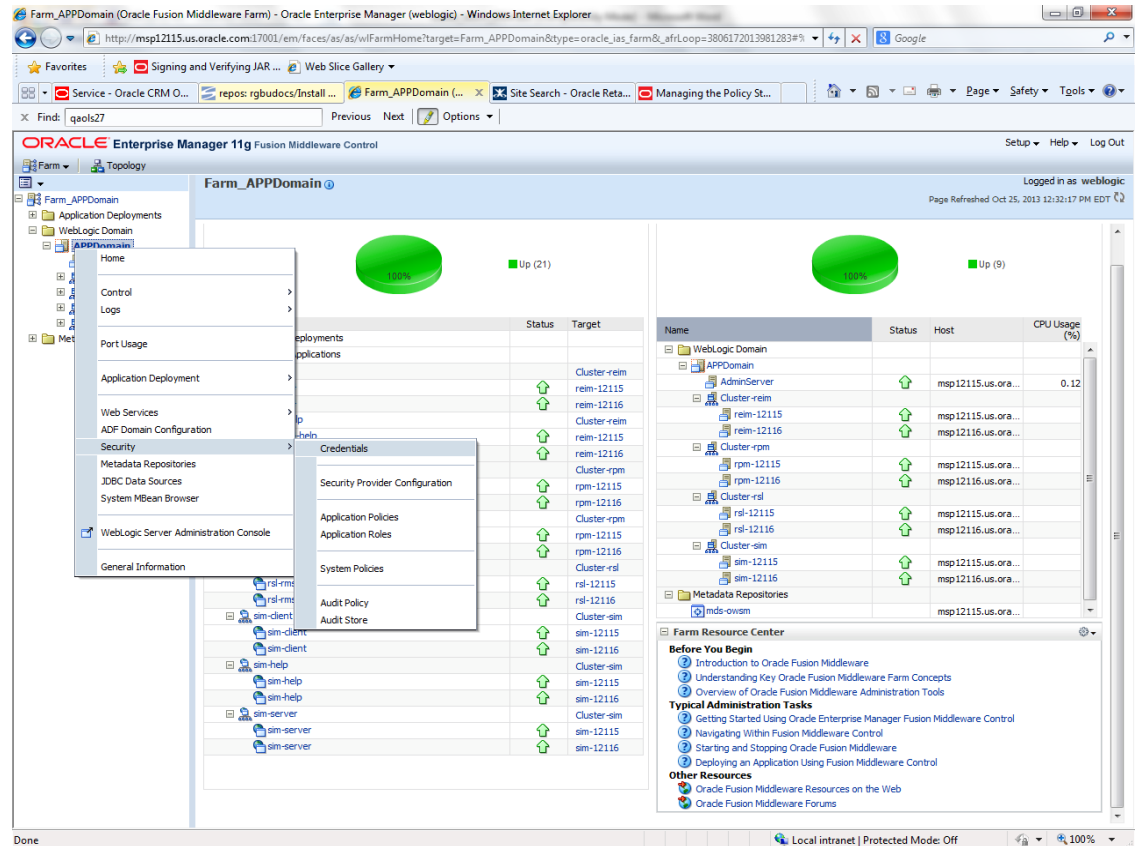
Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser RMS01APP, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments 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 14.1 install guides.

The following sections show an example of how to administer the password stores thru ORACLE Enterprise Manager 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 screenshot shows the Oracle Enterprise Manager 11g Fusion Middleware Control interface. The left-hand navigation pane displays a tree structure with the following items: Farm, Application Deployments, WebLogic Domain, APPDomain (selected), AdminServer, Cluster-rm, Cluster-rpm, Cluster-ral, Cluster-sim, and Metadata Repositories. The main content area is titled 'APPDomain (1)' and 'WebLogic Domain'. It includes a 'Credentials' section with a description: 'A credential store is the repository of security data that certify the authority of entities used by Java 2, J2EE, and ADF applications. Applications can use the Credential Store, a single, consolidated service provider to store and manage their credentials securely.' Below this, the 'Credential Store Provider' is listed as 'Scope: WebLogic Domain' and 'Provider: DO\_ORACLE'. A table of credentials is displayed with columns for 'Credential', 'Type', and 'Description'. The table contains two groups of credentials: 'oracle-retail-sim' and 'jpm14'. The 'oracle-retail-sim' group includes 'bp-user', 'rsi-rpm-user', 'rb-user', 'ral-rms-user', 'server-user', 'delexure-user', 'ldap-user', and 'seo-token-key'. The 'jpm14' group includes 'LDAP-ALIAS', 'user.signature.salt', and 'DE-ALIAS'. The status bar at the bottom indicates 'Done' and 'Local intranet | Protected Mode: Off'.

Oracle Enterprise Manager 11g Fusion Middleware Control

APPDomain (1)

WebLogic Domain

Logged in as weblogic

Page Refreshed Oct 25, 2013 12:49:37 PM EDT

**Credentials**

A credential store is the repository of security data that certify the authority of entities used by Java 2, J2EE, and ADF applications. Applications can use the Credential Store, a single, consolidated service provider to store and manage their credentials securely.

**Credential Store Provider**

Scope: WebLogic Domain  
Provider: DO\_ORACLE

Create Map Create Key Edit... Delete... Credential Key Name

| Credential          | Type     | Description |
|---------------------|----------|-------------|
| oracle-retail-sim   |          |             |
| bp-user             | Password |             |
| rsi-rpm-user        | Password |             |
| rb-user             | Password |             |
| ral-rms-user        | Password |             |
| server-user         | Password |             |
| delexure-user       | Password |             |
| ldap-user           | Password |             |
| seo-token-key       | Generic  |             |
| jpm14               |          |             |
| LDAP-ALIAS          | Password |             |
| user.signature.salt | Password |             |
| DE-ALIAS            | Password |             |

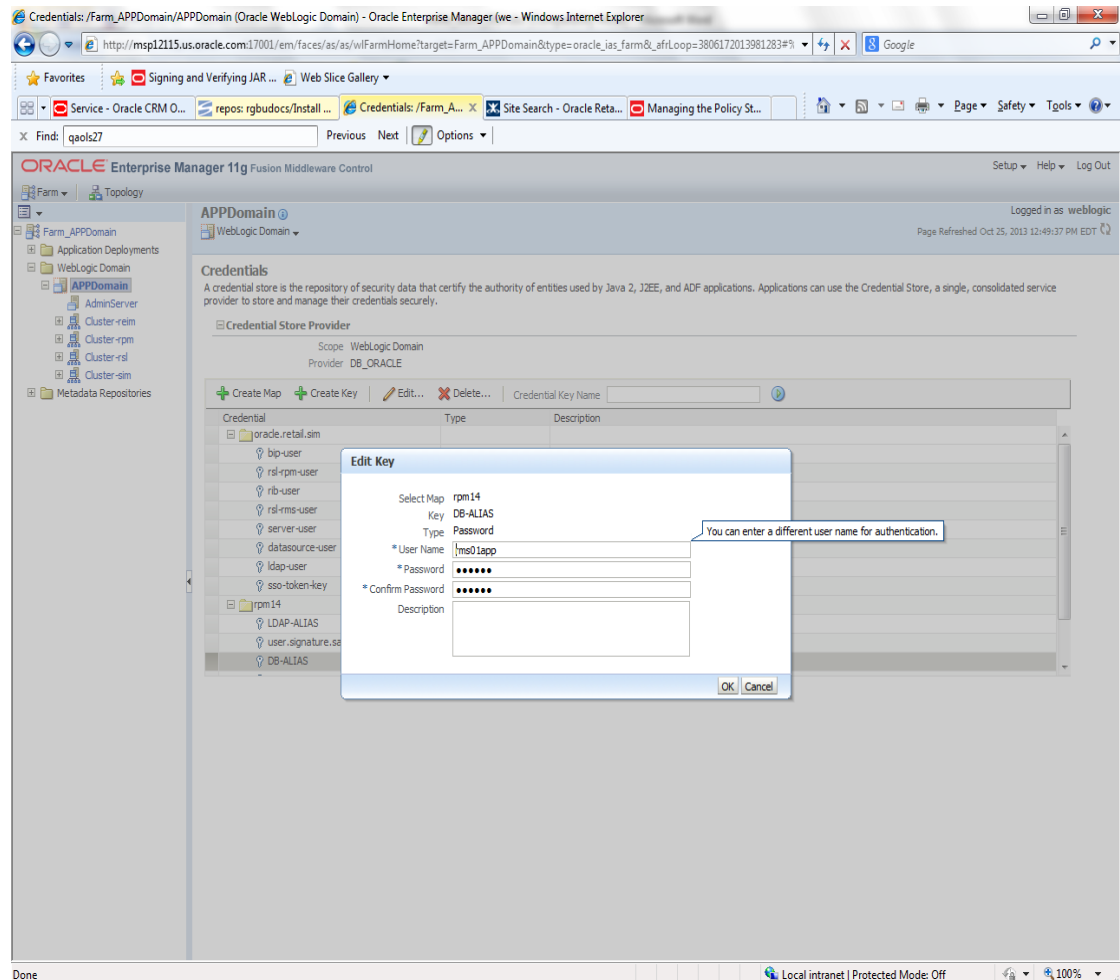
Done

Local intranet | Protected Mode: Off

100%

The Create Map add above is to create a new map with keys under it. A map would usually be an application such as rpm14. 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 (rpm14) 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 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.

For platform-specific requirements to run an OPSS script, see [http://docs.oracle.com/cd/E21764\\_01/core.1111/e10043/managepols.htm#CIHIBBDJ](http://docs.oracle.com/cd/E21764_01/core.1111/e10043/managepols.htm#CIHIBBDJ)

The weakness with the WLST/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 rpm14 and each alias in the wallet:

Oracle PKI Tool : Version 11.1.1.7.0

Requested Certificates:

User Certificates:

Oracle Secret Store entries:

rpm14@#3#@DB-ALIAS

rpm14@#3#@LDAP-ALIAS

rpm14@#3#@RETAIL.USER

rpm14@#3#@user.signature.salt

rpm14@#3#@user.signature.secretkey

rpm14@#3#@WEBLOGIC-ALIAS

rpm14@#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 (WLST)...
```

```
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="rpm14",key="DB-ALIAS")
Already in Domain Runtime Tree
```

```
[Name : rms01app, Description : null, expiry Date : null]
PASSWORD:retail
*The above means for map rpm14 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                         |
|-------------------------------------------|-------------|-----------------------------------------------|--------------------------|-----------------------------------------|------------------------------------|-------------------------------------------|-----------|---------------|-------------------------------|
| <b>RMS batch</b>                          | DB          | <RMS batch install dir (RETAIL_HOME)>/.wallet | n/a                      | <Database SID>_<Data base schema owner> | <rms schema owner>                 | Compile, execution                        | Installer | n/a           | Alias hard-coded by installer |
| <b>RMS forms</b>                          | DB          | <forms install dir>/base/.wallet              | n/a                      | <Database SID>_<Data base schema owner> | <rms schema owner>                 | Compile                                   | Installer | n/a           | Alias hard-coded by installer |
| <b>ARI forms</b>                          | DB          | <forms install dir>/base/.wallet              | n/a                      | <Db_Ari01>                              | <ari schema owner>                 | Compile                                   | Manual    | ari-alias     |                               |
| <b>RMWS forms</b>                         | DB          | <forms install dir>/base/.wallet              | n/a                      | <Database SID>_<Data base schema owner> | <rwms schema owner>                | Compile forms, execute batch              | Installer | n/a           | Alias hard-coded by installer |
| <b>RPM batch<br/>plsql and<br/>sqlldr</b> | 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  |
| <b>RWMS auto-<br/>login</b>               | JAVA        | <forms install dir>/base/.javawallet          |                          |                                         |                                    |                                           |           |               |                               |
|                                           |             |                                               | <RWMS Installation name> | <RWMS database user alias>              | <RWMS schema owner>                | RWMS forms app to avoid dblogin screen    | Installer | rwms14inst    |                               |
|                                           |             |                                               | <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                                                                                                                                          |
|----------------|---------------------|--------------------------------------------------------------|-------------------------------------------------------|----------------------------------|---------------------------------|---------|-----------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AIP app</b> | JAVA                | <weblogic domain home>/retail/<deployed aip app name>/config |                                                       |                                  |                                 |         |           |                        | Each alias must be unique                                                                                                                      |
|                |                     |                                                              | aip14                                                 | <AIP weblogic user alias>        | <AIP weblogic user name>        | App use | Installer | aip-weblogic-alias     |                                                                                                                                                |
|                |                     |                                                              | aip14                                                 | <AIP database schema user alias> | <AIP database schema user name> | App use | Installer | aip01user-alias        |                                                                                                                                                |
|                |                     |                                                              | aip14                                                 | <rib-aip weblogic user alias>    | <rib-aip weblogic user name>    | App use | Installer | rib-aip-weblogic-alias |                                                                                                                                                |
| <b>RPM app</b> | DB credential store |                                                              | Map=rpml4 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. |
| <b>RPM app</b> | JAVA                | <weblogic domain home>/retail/<deployed rpm app name>/config |                                                       |                                  |                                 |         |           |                        | Each alias must be unique                                                                                                                      |
|                |                     |                                                              | rpm14                                                 | <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                     |
|-----------------|-------------|---------------------------------------------------------------|----------------------------------|----------------------------------------------------------------------|----------------------------------------------|----------------|-----------|-----------------|---------------------------|
|                 |             |                                                               | rpm14                            | <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      |                           |
| <b>ReIM app</b> | JAVA        | <weblogic domain home>/retail/<deployed reim app name>/config |                                  |                                                                      |                                              |                |           |                 | Each alias must be unique |
|                 |             |                                                               | <installed app name, ex: reim14> | <reim weblogic user alias>                                           | <reim weblogic user name>                    | App use        | Installer | weblogic-alias  |                           |
|                 |             |                                                               | <installed app name, ex: reim14> | <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: reim14> | <reim webservice validation user alias> | <reim webservice validation user name>        | App use        | Installer | reimwebser vice-alias |                           |
|            |             |                                         | <installed app name, ex: reim14> | <reim batch user alias>                 | <reim batch user name>                        | App, batch use | Installer | reimbat-alias         |                           |
|            |             |                                         | <installed app name, ex: reim14> | <LDAP-ALIAS>                            | cn=REIM.ADMIN,cn=Users,dc=us,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.ADMIN,cn=Users,dc=us,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                                                                                                                                                                                                                       |
|-----------------|---------------------|---------------------------------------------------------------|-------------------------------------------------------|-----------------------------|---------------------------|---------|-----------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RESA app</b> | DB credential store |                                                               | Map=resa14 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. |
| <b>RESA app</b> | 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/conf/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<br>weblogic<br>user alias>  | <resa<br>weblogic<br>user name>  | App use | Installer | wlsalias       |                                                                                                                                                                                                                             |
|                  |                     |                                                | retail_installer                                        | <resa<br>schema db<br>user alias> | <rmsdb<br>shema user<br>name>    | App use | Installer | Resadb-alias   |                                                                                                                                                                                                                             |
|                  | JAVA                | <retail_<br>home>/orpatch/config/javaapp_rasrm |                                                         |                                   |                                  |         |           |                | Each alias must be unique                                                                                                                                                                                                   |
|                  |                     |                                                | retail_installer                                        | <alloc<br>weblogic<br>user alias> | <alloc<br>weblogic<br>user name> | App use | Installer | weblogic-alias |                                                                                                                                                                                                                             |
| <b>Alloc app</b> | DB credential store |                                                | Map=alloc 14 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. |
| <b>Alloc app</b> | 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                                                                                                                                          |
|----------------|---------------------|-------------------------------------------------------------------------------|-----------------------|----------------------------------|--------------------------------|-----------------|-----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SIM app</b> | 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      |                                                                                                                                                |
| <b>RETL</b>    | 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                                                                                      |
| <b>RETL</b>    | 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                                                                                      |
| <b>RIB</b>     | JAVA                | <RIBHOME DIR>/deployment-home/conf/security                                   |                       |                                  |                                |                 |           |                     | <app> is one of aip, rfm, rms, rpm, sim, rwms, tafr                                                                                            |
| <b>JMS</b>     |                     |                                                                               | 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                               |
|-----------------------|-------------|------------------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------|----------------------------------------------------|-----------------|-----------|----------------------------------|-------------------------------------|
| <b>WebLogic</b>       |             |                                                                                          | rib-<app>-app-server-instance | <rib-app weblogic user alias>                       | <rib-app weblogic user name>                       | Integration use | Installer | weblogic-alias                   |                                     |
| <b>Admin GUI</b>      |             |                                                                                          | rib-<app>#web-app-user-alias  | <rib-app admin gui user alias>                      | <rib-app admin gui user name>                      | Integration use | Installer | admin-gui-alias                  |                                     |
| <b>Application</b>    |             |                                                                                          | rib-<app>#user-alias          | <app weblogic user alias>                           | <app weblogic user name>                           | Integration use | Installer | app-user-alias                   | Valid only for aip, rpm, sim        |
| <b>DB</b>             |             |                                                                                          | 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 |
| <b>Error Hospital</b> |             |                                                                                          | 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                  |                                     |
| <b>RFI</b>            | 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: 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 Agent (WebGate), used to authenticate the user and create the Single Sign-On cookies.
- Oracle Directory Services Manager (ODSM) application in OIM11g, 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.

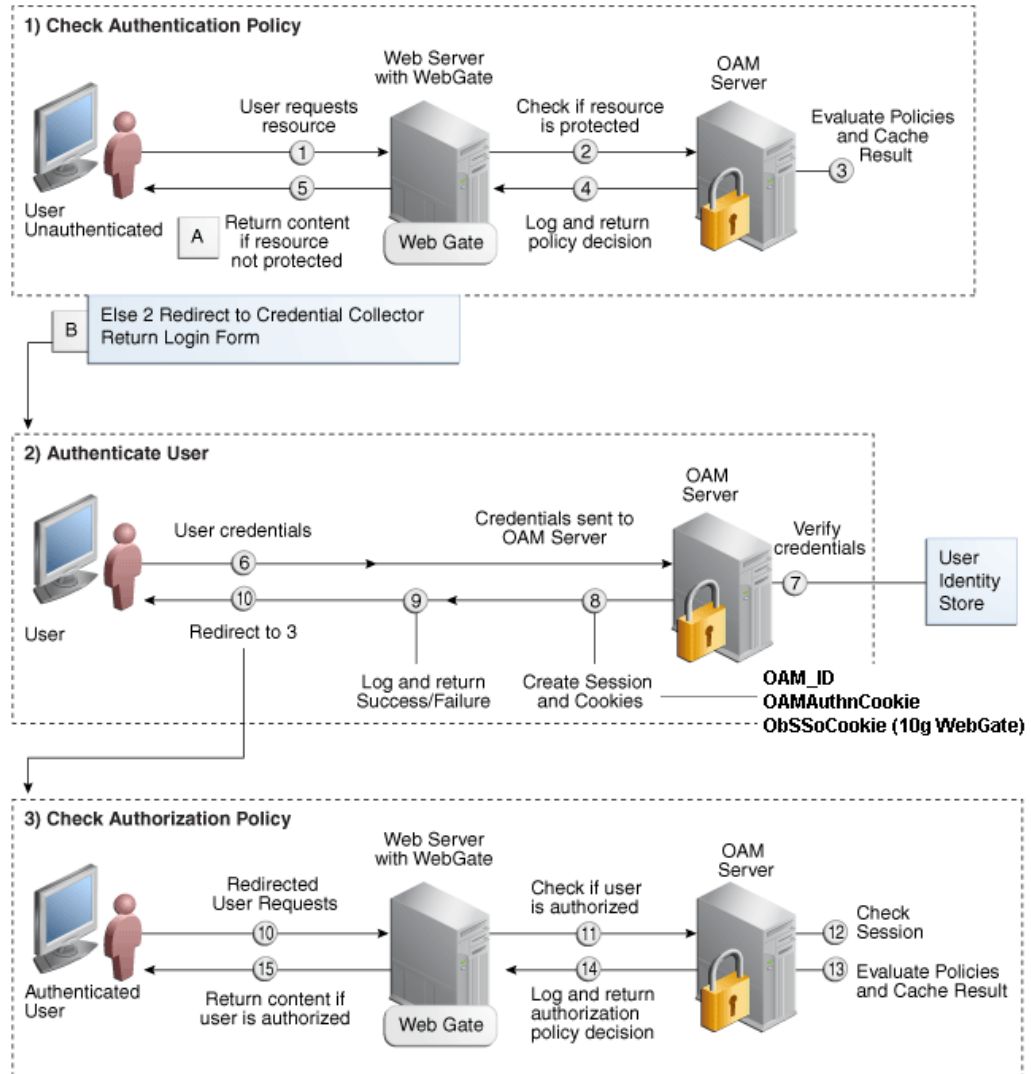
### About SSO Login Processing with OAM Agents

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: 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)

---

---

**Note:** During installation of RPM, you are asked for the RIBforRPM provider URL. Because RIB is installed after RPM, make a note of the URL you enter. To change the RIBforRPM provider URL after you install RIB, edit the `remote_service_locator_info_ribserver.xml` file.

---

---

8. Oracle Retail Allocation
9. Oracle Retail Mobile Merchandising (ORMM)
10. Oracle Retail Central Office (ORCO)
11. Oracle Retail Returns Management (ORRM)
12. Oracle Retail Back Office (ORBO)
13. Oracle Retail Store Inventory Management (SIM)

---

---

**Note:** During installation of SIM, you are asked for the RIB provider URL. Because RIB is installed after SIM, make a note of the URL you enter. To change the RIB provider URL after you install RIB, edit the `remote_service_locator_info_ribserver.xml` file.

---

---

14. Oracle Retail Predictive Application Server (RPAS)
15. Oracle Retail Demand Forecasting (RDF)
16. Oracle Retail Category Management (RCM)
17. Oracle Retail Replenishment Optimization (RO)
18. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
19. Oracle Retail Regular Price Optimization (RPO)
20. Oracle Retail Merchandise Financial Planning (MFP)
21. Oracle Retail Size Profile Optimization (SPO)

22. Oracle Retail Assortment Planning (AP)
23. Oracle Retail Item Planning (IP)
24. Oracle Retail Item Planning Configured for COE (IP COE)
25. Oracle Retail Advanced Inventory Planning (AIP)
26. Oracle Retail Analytics
27. Oracle Retail Advanced Science Engine (ORASE)
28. Oracle Retail Integration Bus (RIB)
29. Oracle Retail Service Backbone (RSB)
30. Oracle Retail Financial Integration (ORFI)
31. Oracle Retail Point-of-Service (ORPOS)
  - Oracle Retail Mobile Point-of-Service (ORMPOS) (requires ORPOS)
32. Oracle Retail Markdown Optimization (MDO)
33. Oracle Retail Clearance Optimization Engine (COE)
34. Oracle Retail Analytic Parameter Calculator for Markdown Optimization (APC-MDO)
35. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
36. Oracle Retail Macro Space Planning (MSP)

The Oracle Retail Enterprise suite includes Macro Space Planning. This can be installed independently of and does not affect the installation order of the other applications in the suite. If Macro Space Planning is installed, the installation order for its component parts is:

  - Oracle Retail Macro Space Management (MSM)
  - Oracle Retail In-Store Space Collaboration (ISSC) (requires MSM)
  - Oracle Retail Mobile In-Store Space Collaboration (requires MSM and ISSC)