Oracle® Retail Price Management

Installation Guide
Release 13.2.6
E47693-02

July 2013

ORACLE

Oracle® Retail Price Management Installation Guide, Release 13.2.6

Primary Author: Wade Schwarz

Contributors: Nathan Young and Kelly Baranick

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

Oracle Retail Price Management Installation Guide, Release 13.2.6

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 13.2.6 documentation set:

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

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, 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, 13.2) or a later patch release (for example, 13.2.6). If you are installing the base release and additional patch and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:
http://www.oracle.com/technology/documentation/oracle_retail.html

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

Documentation should be available on this Web site within a month after a product release.

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
```

1

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.

> **Note:** Oracle Retail product installations are tightly integrated with their technical configuration. After installation the application server hostname, database name and hostname, and other technical configuration is embedded within the installation of the Oracle Retail product. It is not recommended to attempt to copy an installation to a server with a different hostname for the purposes of environment cloning. The easiest and safest way to reconfigure applications on another server is to reinstall the applications using the Oracle Retail installers.

Patch Contents

Patch releases include all defect fixes that have been released through bundled hot fix releases since the last patch release. Patch releases may also include new defect fixes and enhancements that have not previously been included in any bundled hot fix release.

The *Oracle Retail Price Management 13.2.3.1 Release Notes* contained incorrect bundled hot fix installation procedures for WebLogic. See the *Oracle Retail Price Management Corrected Bundled Hot Fix Installation on WebLogic* (My Oracle Support Note 1473368.1) for the updated instructions.

Check for the Current Version of the Installation Guide

Corrected versions of Oracle Retail installation guides may be published whenever critical corrections are required. For critical corrections, the rerelease of an installation guide may not be attached to a release; the document will simply be replaced on the Oracle Technology Network Web site.

Before you begin installation, check to be sure that you have the most recent version of this installation guide. Oracle Retail installation guides are available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technology/documentation/oracle_retail.html

An updated version of an installation guide is indicated by 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 an installation guide with part number E123456-**01**.

If a more recent version of this installation guide is available, that version supersedes all previous versions. Only use the newest version for your installation.

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 11gR2 Enterprise Edition. Options are: ▪ Oracle Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 6.1 (Actual hardware or LPARs) ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 10 SPARC (Actual hardware or logical domains) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.3) with the following specifications: **Components:** ▪ Oracle Partitioning ▪ Examples CD (Formerly the companion CD) **Oneoff Patches:** ▪ 13036331: ORA-01031 INSUFFICIENT PRIVILEGES WHEN GRANTING EXECUTE ON DBMS PACKAGES. **Other components:** ▪ Perl 5 or later ▪ X-Windows interface

Check Supported Application Server Requirements

General requirements for an application server capable of running the Oracle Retail Price Management application include the following.

Note: Files required for OCM (Oracle Configuration Manager) are removed after OPatch is used to patch the WebLogic server. This will cause the product installers and OCM installation to fail. To work around this issue, back up the content of the $ORACLE_HOME/utils/ccr/lib directory prior to applying a patch using OPatch, and recopy the content back after you apply any patches. ORACLE_HOME is the location where WebLogic Server has been installed.

Note: If using an OPatch on Linux 64-bit platforms, see Installer Fails because of missing .jar in $ORACLE_HOME/utils/ccr/lib in Appendix: Common Installation Errors.

Supported on	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.6). Options are: ▪ Oracle Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 6.1 (Actual hardware or LPARs) ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 10 SPARC (Actual hardware or logical domains) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)

Supported on	Versions Supported
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.6) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.6) • Java: JDK 1.6.0+ 64 bit or JDK 1.7.0+ 64 bit or Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only. **Optional (SSO required)** ▪ Oracle WebTier 11g (11.1.1.6) ▪ Oracle Internet Directory 10gR3 (10.1.4) optionally with Oracle Single Sign-On 10gR3 (10.1.4) or ▪ Oracle Identity Management 11gR1 (11.1.1.6) optionally with Oracle Single Sign-On 10gR3 (10.1.4) or ▪ Oracle Identity Management 11gR1 (11.1.1.6) optionally with Oracle Access Manager 11gR1 (11.1.1.5). Must have separate WebLogic 10.3.5 for Oracle Access Manager 11g. **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 11.1.1.6. Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed. **Note:** See Installers fail because of missing .jar file in $ORACLE_HOME/utils/ccr/lib in" Appendix: Common Installation Errors." This issue occurs only when the application is being installed on the same WebLogic server where forms based applications are installed. It is valid only for Linux 64-bit.

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 Internet Directory 10gR3 version 10.1.4 or Oracle Identity Management 11gR1 version 11.1.1.6 has been installed along with the components listed in the above Application Server requirements section. Verify the Oracle WebTier Server is registered with the Oracle Access Manager 11gR1 as a partner application.

Check Supported Client PC and Web Browser Requirements

Requirement	Version
Operating system	Windows XP or Windows 7
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	1.6.0_22+ or 1.7+
Browser	Microsoft Internet Explorer version 8.0 or 9.0 Mozilla Firefox 3.6 or 10.0 or Mozilla Firefox ESR 17.0.3+ **Note:** Other Oracle Merchandising applications may not have the same levels of browser certification.

Check Oracle Retail Software Dependencies

RMS application database portion 13.2.6 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)	13.2.6
Oracle Retail Allocation	13.2.6 or 13.3
Oracle Retail Store Inventory Management (SIM)	13.2.6
Oracle Retail POS Suite	13.3.6 or 13.4.5

Supported Oracle Retail Integration Technologies

Requirement	Version
Oracle Retail Integration Bus (RIB)	13.2.6
Oracle Retail Service Layer (RSL)	13.2.6

Check Third-Party Software Dependencies

Hibernate 2.1.8 must be downloaded and the hibernate2.jar file just be extracted. The RPM application installation procedure specifies how to install this file.

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.

2

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 an 11.2.0.3 RAC database. When using a RAC database, all JDBC connections should be configured to use THIN connections rather than OCI connections. It is suggested that if you do use OCI connections, the Oracle Retail products database be configured in the tnsnames.ora file used by the WebLogic Server installations.

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 11.2.0.3 Oracle Internet Directory database with the WebLogic 10.3.6 cluster. It is suggested that a Web Tier 11.1.1.6 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 11g Release 2 (11.2) Part Number E16795-11

3

Database Installation Tasks

RPM Schema

The RPM database tables are installed with the RMS database schema. RMS 13.2.6 is a prerequisite of the RPM 13.2.6 installation.

4

Application Installation Tasks

Before proceeding, you must install Oracle WebLogic Server 11g Release 1 (10.3.6) and the patches listed in Chapter 1, "Preinstallation Tasks."

> **IMPORTANT**: If there is an existing WebLogic installation on the server, you must upgrade to WebLogic 10.3.6. All middleware components associated with WebLogic server should be upgraded to 11.1.1.6.
>
> Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.

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 RMS and Oracle Retail Price Management schemas for your installation.

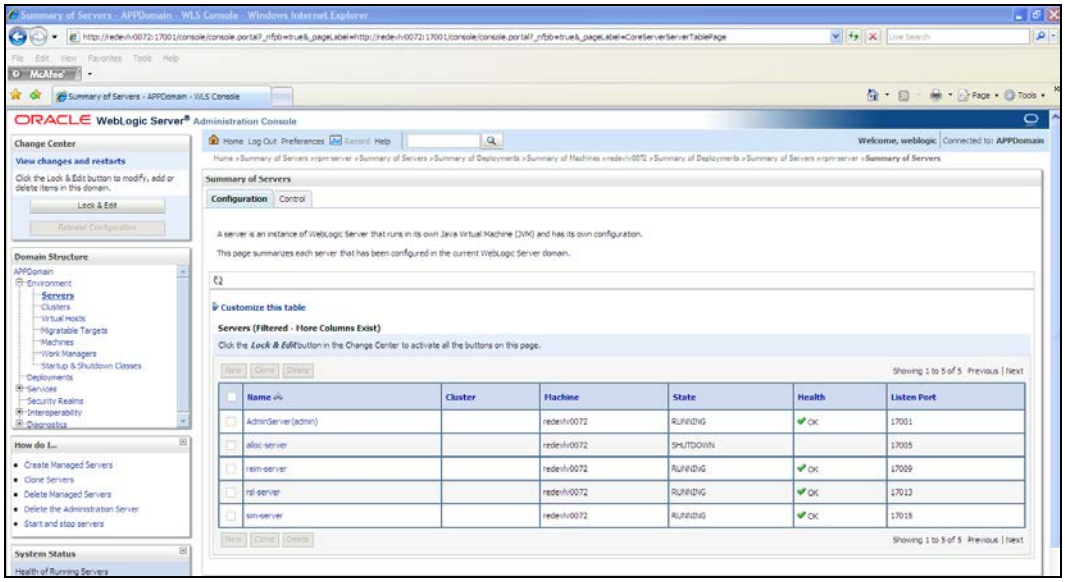
If Oracle Forms 11g has been installed in the same WebLogic being used for this application, a domain called ClassicDomain is installed. Installing a separate domain under the same WebLogic server is recommended. It can be called APPDomain (or something similar) and will be used to install the non-ORACLE Forms managed servers. Applications such as RPM, SIM, Allocation, ReIM, RIB, AIP, and RSL can be installed in the APPDomain.

Install Managed Server in WebLogic

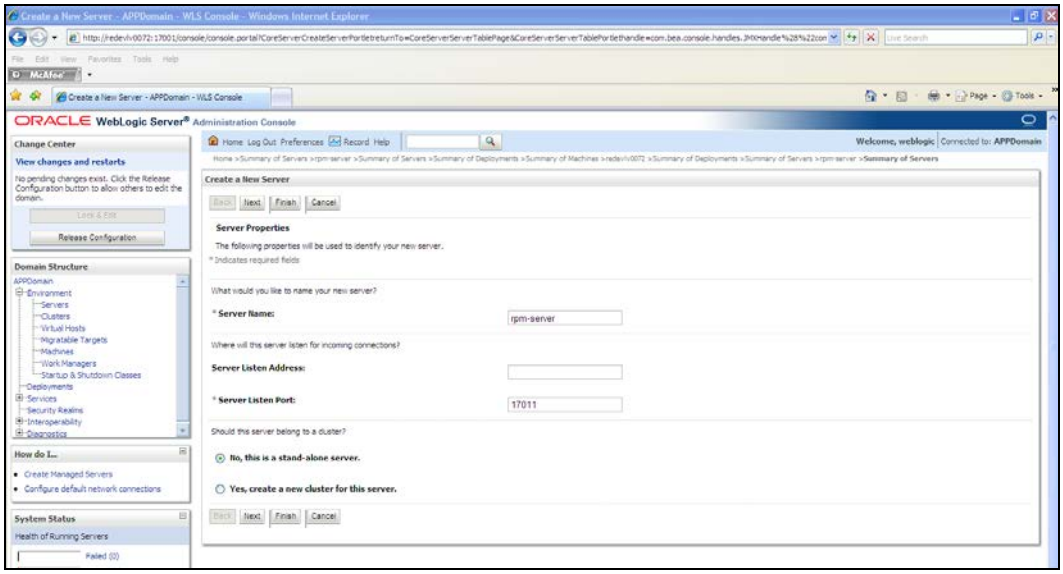
> **Important Note:** Skip this section if a managed server already exists for RPM.

Before running the application installer, you must install the managed server in WebLogic if it was not created during the domain install.

1. Log in to the Administration Console.



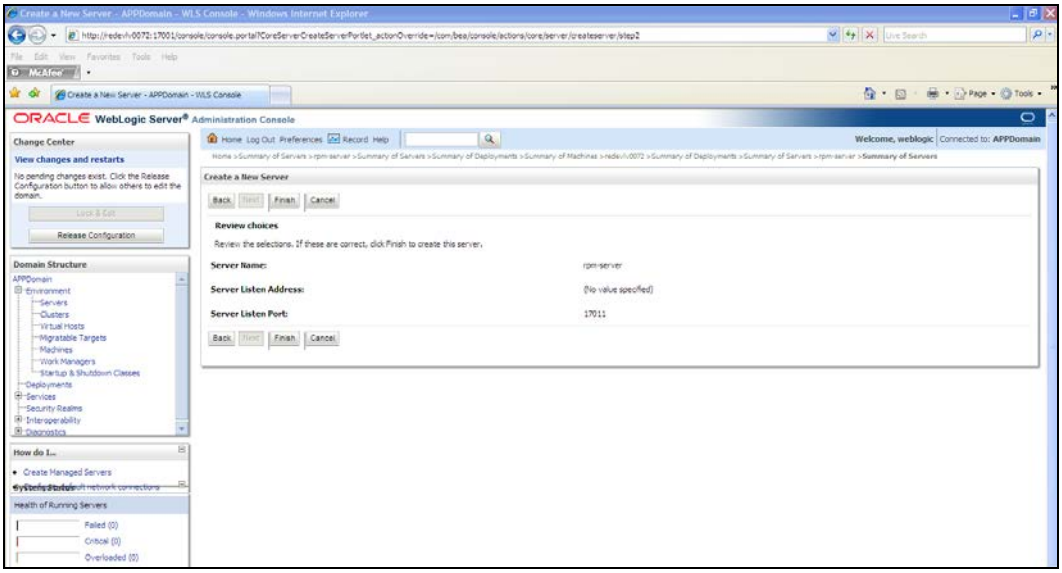
2. Click **Lock & Edit**.
3. Navigate to Environment> Servers and select new tab of the servers on the right side.



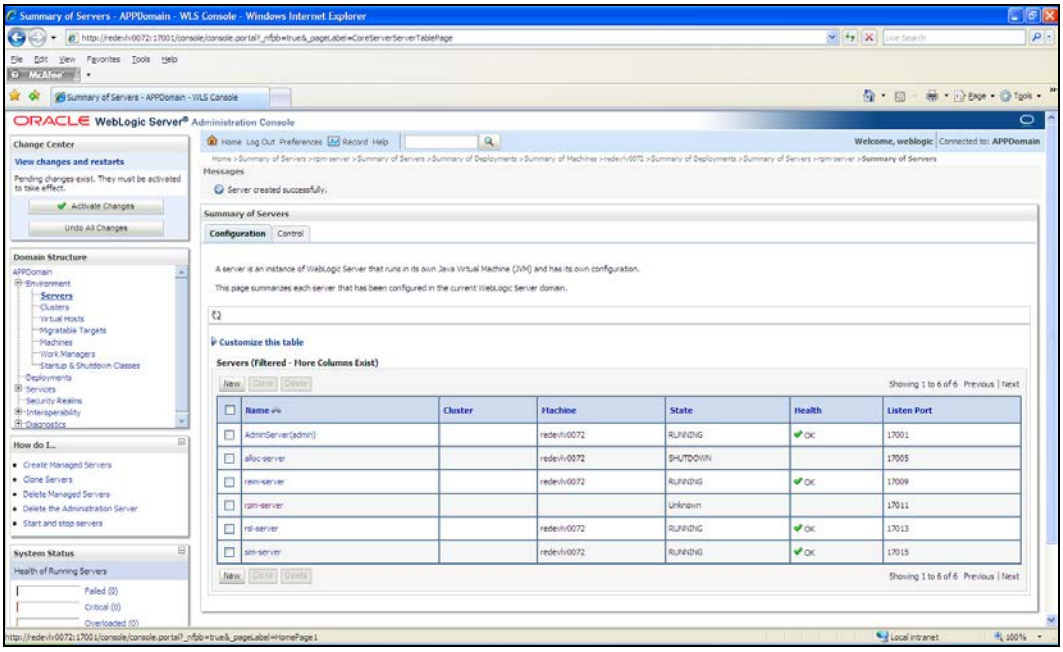
4. Set the following variables.
   - **Server Name**: These should be some name specific to your application targeted (for example, rpm-server).
   - **Server Listen Address**: <weblogic server> (for example, redevlv0072.us.oracle.com)
   - **Server Listen Port**: Availableport; you should check for availability.

     A suggestion is to increment the AdminServer port by two and keep incrementing by two for each managed server (for example 17007, 17009, 170011, and so on.)

5. Click **Next**.



6. Click **Finish**.

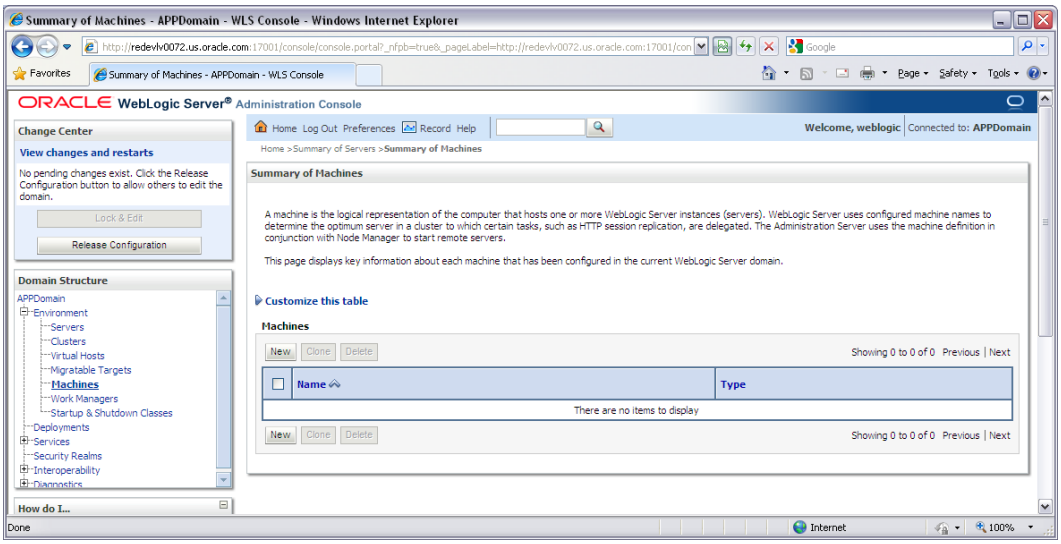


7. Click **Activate Changes** on the left side.

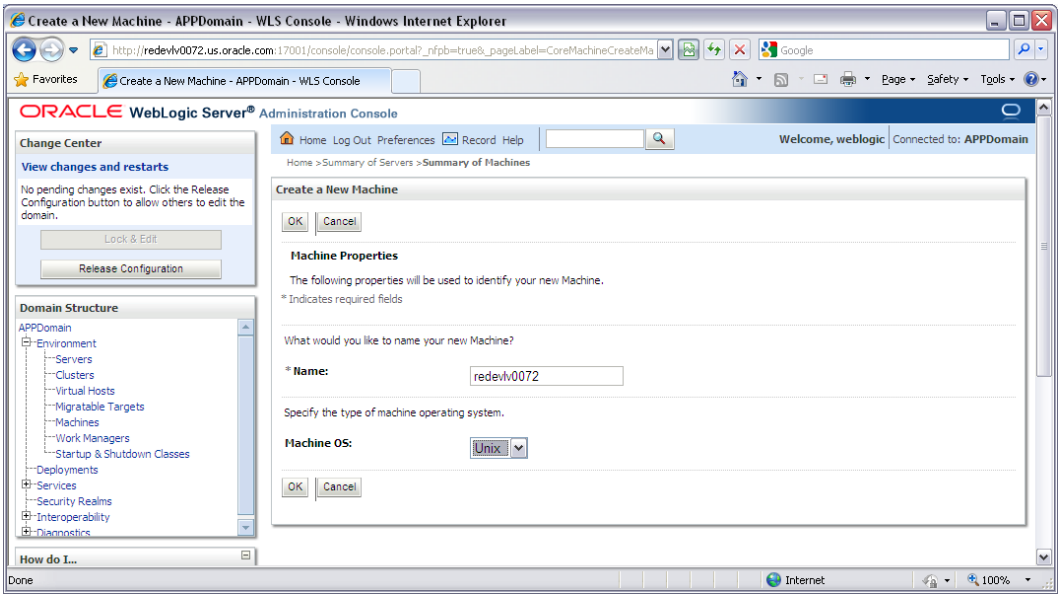
Install NodeManager

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

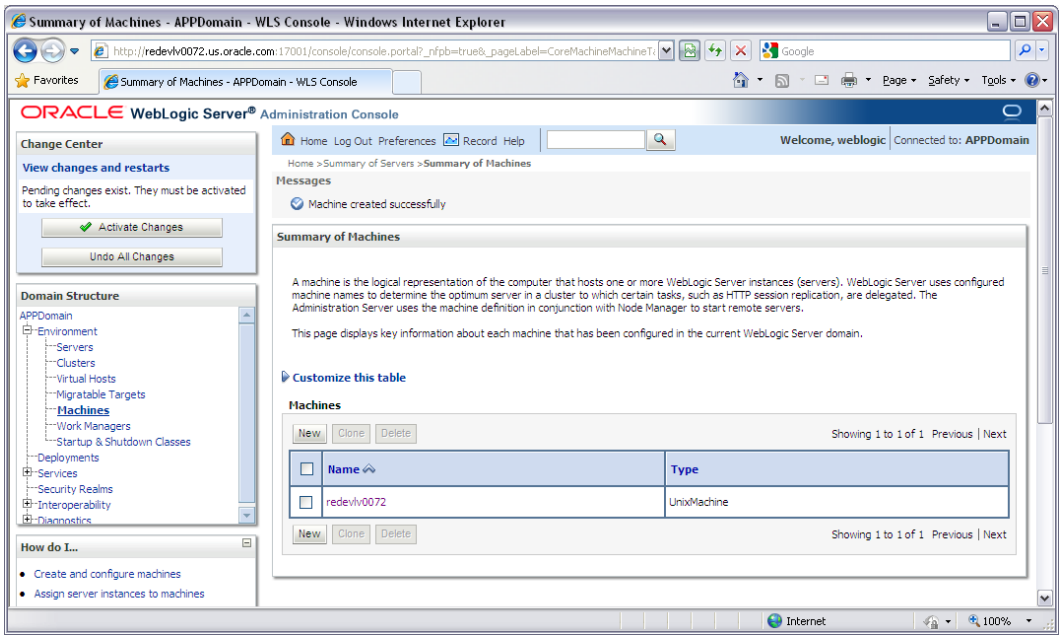
1. Log in to the Administration Console.
2. Click **Lock & Edit** and navigate to Environments->Machines.



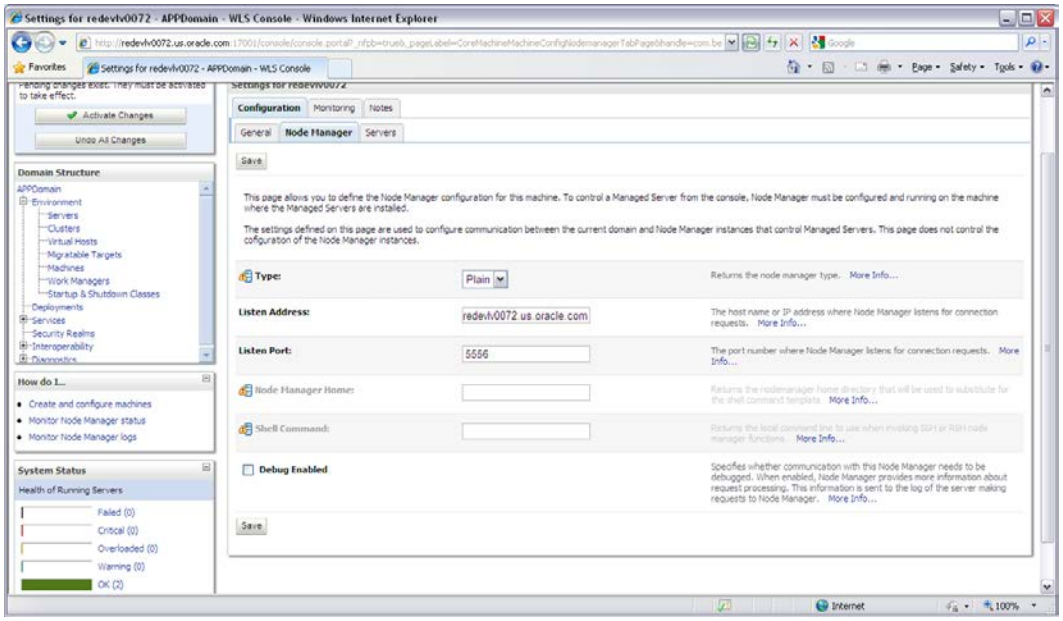
3. Click **New**.



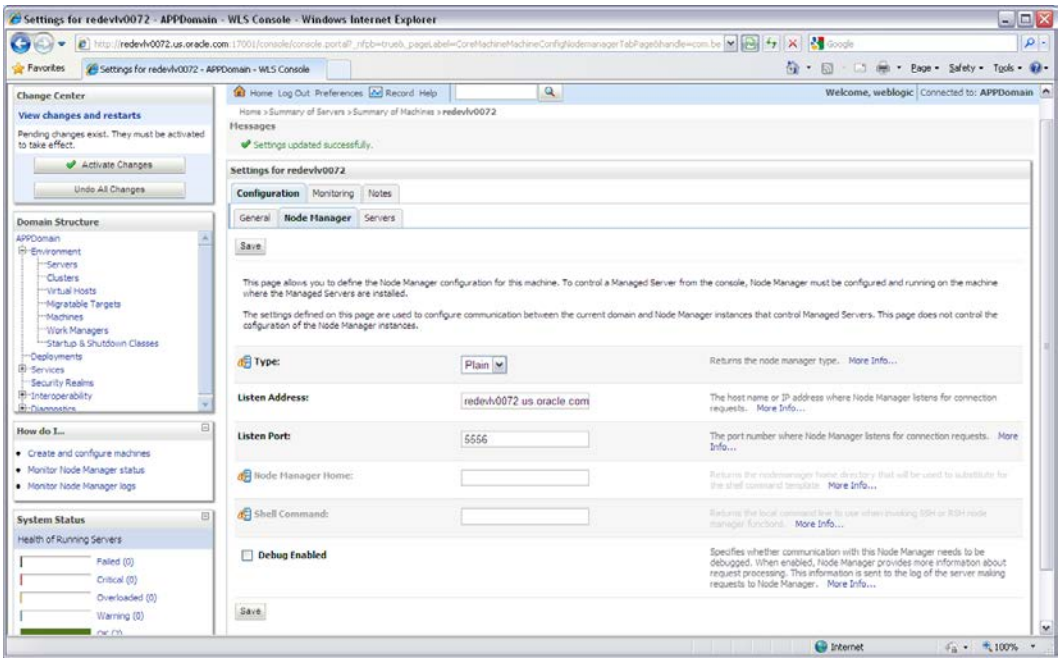
4. Set the following variables:
   - **Name**: Logical machine name
   - **Machine OS**: UNIX
5. Click **OK**.
6. Click on the machine created below.



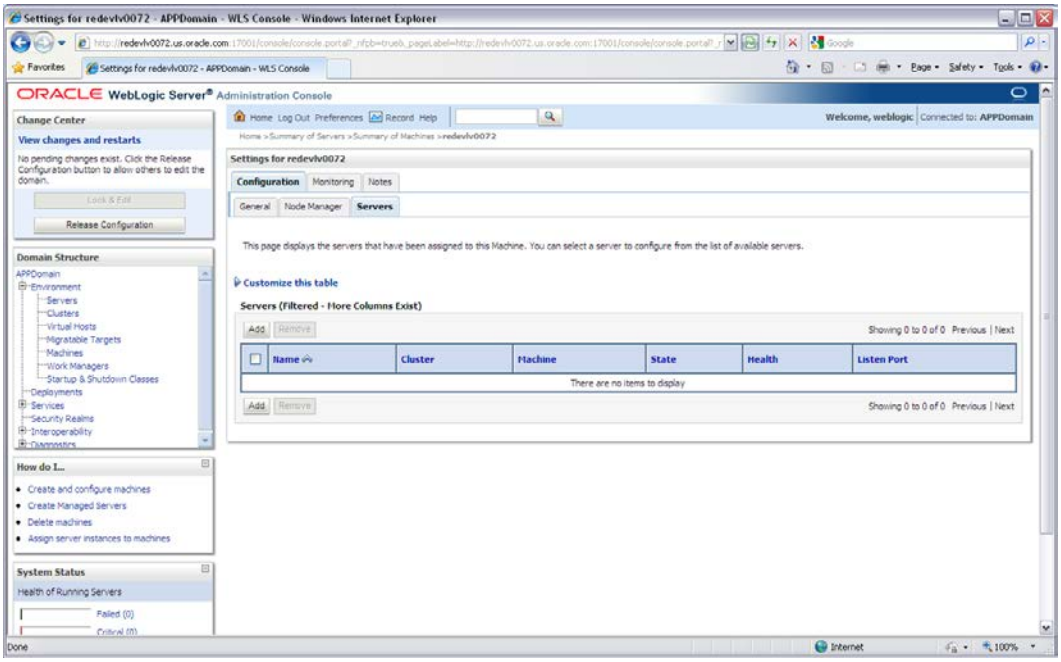
7. Click on the NodeManager tab and update the details below.
   - **Type**: Plain
   - **Listen Address**: redevlv0072.us.oracle.com
   - **Listen Port**: default port (for example, 5556) or any available port.



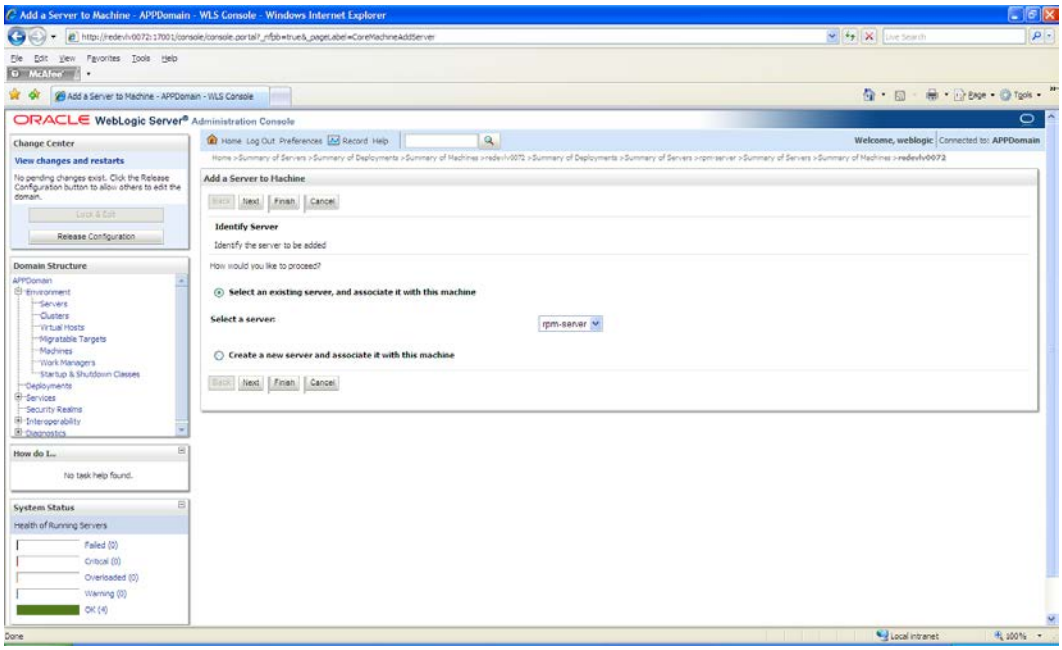
8. Click **Save**.



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



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



13. Set the following variables:
    - Server: rpm-server
14. Click **Next/Finish**.
15. Click **Activate Changes**.

> **Note:** To activate changes, the server must be stopped:
>
> $WLS_HOME>/user_projects/domains/<APP_Domain>/bin/stopManagedWebLogic.sh <rpm>-server ${server_name}:${server_port}

16. Start NodeManager from the server using the startNodeManager.sh at $WLS_HOME/wlserver_10.3/server/bin.
17. Edit the nodemanager.properties file at the following location with the below values:

    $WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties
    - SecureListener=false
    - StartScriptEnabled=true
    - StartScriptName=startWebLogic.sh.
18. 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 Managed Servers

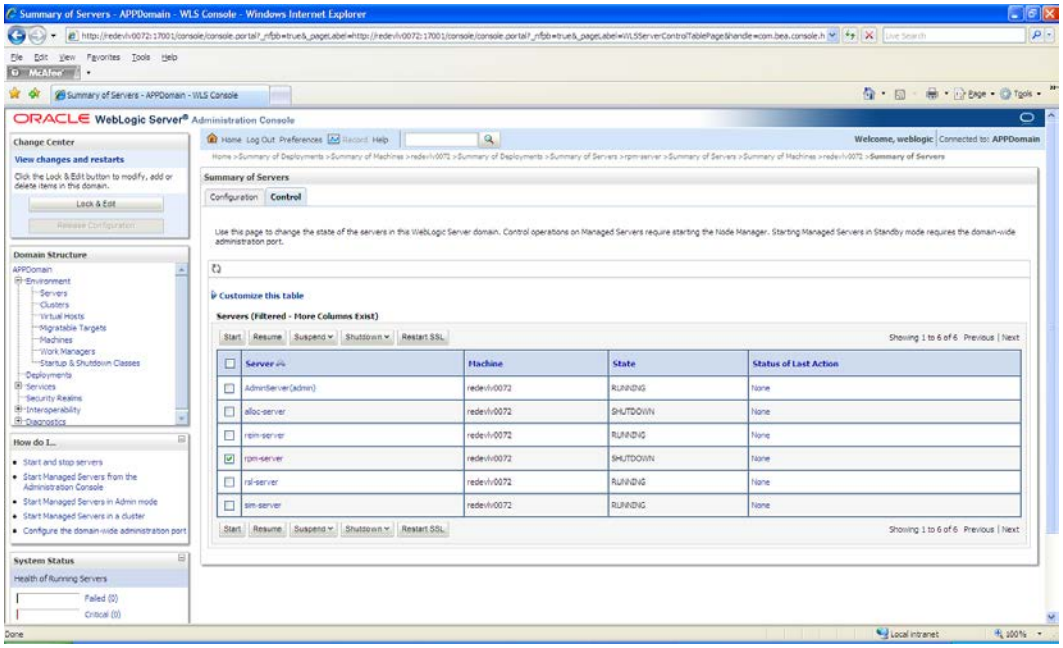
To start the managed servers, complete the following steps.

1. Start NodeManager from the command line.

   `$WLS_HOME/wlserver_10.3/server/bin/startNodeManager.sh`

   After NodeManager is started, the managed servers can be started through the Administration Console.

2. Navigate to Environments->Servers->select <app>-server managed server and click the Control tab.



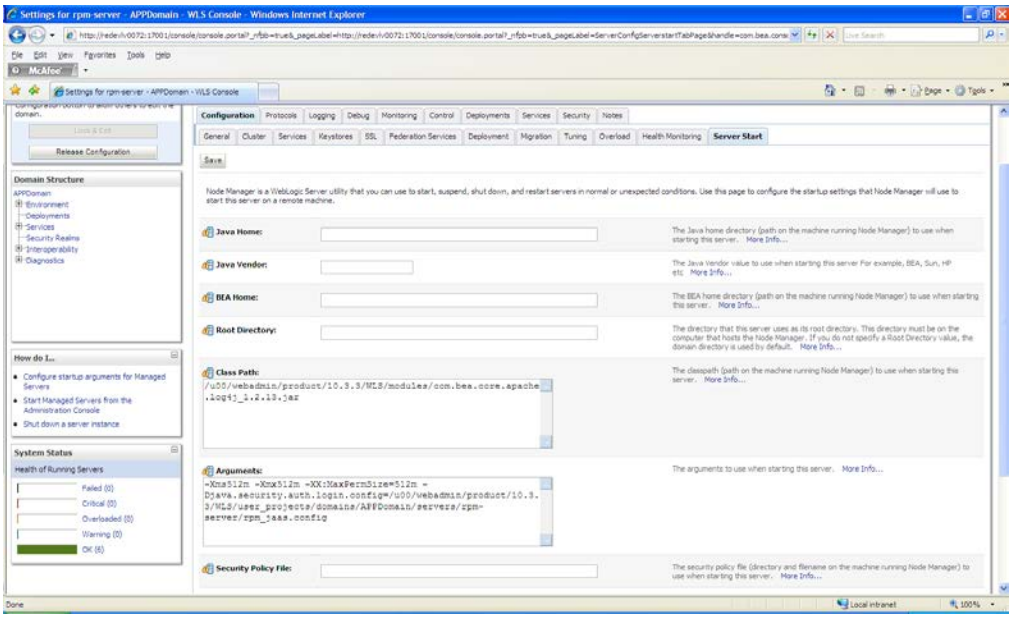
Note: The following arguments are required when starting this managed server using scripts outside of WebLogic console.

Arguments for 1.6.0+ JDK

```
-Xms512m -Xmx512m -XX:MaxPermSize=512m -
Djava.security.auth.login.config=<WLS_HOME>/user_projects/domains/<domain_name
>/servers/<rpm managedservername>/rpm_jaas.config
```

Arguments for Jrockit

```
-Xms512m -Xmx512m -
Djava.security.auth.login.config=<WLS_HOME>/user_projects/domains/<domain_name
>/servers/<rpm managedservername>/rpm_jaas.config
```



> **Note:** Typically, rpm_jaas.config is found in <WEBLOGIC_DOMAIN_HOME>/servers/<rpm-managed-server>. It may be in a different path for cluster environment. You must validate the path of rpm_jaas.config before saving the changes. This file will not exist until after installer has been successfully run.

3. Export WEBLOGIC_DOMAIN_HOME=<WLS_HOME>/user_projects/domains/<domain name>
4. 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 semicolon.

> **Note:** <WEBLOGIC_DOMAIN_HOME> in the below example is the full path of the WebLogic Domain, <managed_server> is the RPM managed server created and <context_root> correlates to the value entered for the application deployment name/context root of the application that you will supply during installation. See the example. There should not be any space between file:<WEBLOGIC_DOMAIN_HOME.

```
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/10.3.x/WLS/user_projects/domains/APPDomain/servers/rpm
-server/tmp/_WL_user/rpm13/-" {
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/10.3.x/WLS/user_projects/domains/APPDomain/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";
};
```

5. Restart WebLogic admin server after making changes to the weblogic.policy file in the previous step.

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 (rpm13application.zip). There should be a minimum of 770 MB 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 rpm13application.zip to STAGING_DIR and extract its contents.

Provide the Hibernate Jar File

The RPM application requires the hibernate2.jar file to be installed. This file should be downloaded from http://www.hibernate.org and placed in the STAGING_DIR/rpm/application/hibernate folder before the installer is launched. For RPM 13, Hibernate 2.1.8 should be used. You need to download the Hibernate distribution and extract the hibernate2.jar file from it.

The RPM application installer verifies that hibernate2.jar has been provided and that it is the correct version. If hibernate2.jar is missing or incorrect, the installer does not proceed.

The installer applies hibernate2.jar to the RPM application by placing it under the STAGING_DIR/rpm/application/hibernate.

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:
   input.jms.module = rpmJMSModule

   input.taskqueue.name = taskQueue

   input.chunkqueue.name = chunkQueue
3. Insert into $WEBLOGIC_HOME/wlserver_10.3/server/lib/weblogic.policy file, the same RPM entries for java security permissions you entered on the main server. See the "Start the Managed Servers" section for additional information.

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 6.0 (1.6.0) JDK (or Jrockit 1.6 Build R 28 or higher within the 1.6 code line for Linux or Solaris OS). JDK. WEBLOGIC_DOMAIN_HOME should point to your WebLogic domain.
3. 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.
4. Run the install.sh script. This launches the installer. After installation is complete, a detailed installation log file is created (rpm13install.<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.

Oracle Configuration Manager

The Oracle Retail OCM Installer packaged with this release installs the latest version of OCM.

The following document is available through My Oracle Support Access:
Oracle Configuration Manager Installer Guide (ID 1071030.1)

My Oracle Support is at the following URL:

`https://support.oracle.com`

This guide describes the procedures and interface of the Oracle Retail Oracle Configuration Manager Installer that a retailer runs at the beginning of the installation process.

OCM Documentation Link

`http://www.oracle.com/technology/documentation/ocm.html`

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.

Review or Configure Oracle Single Sign-On

Skip this section if you are not using Single Sign-On for user identification and authentication.

Single Sign-On is applicable only to the JnlpLaunch Servlet. The JnlpLaunch Servlet is a dynamically protected application. The JnlpLaunch Servlet causes the RPM client application to execute under the SSO user name with a temporary password.

> **Note:** The JnlpLaunch servlet may be configured for either an SSO or non-SSO environment.

HTTP Server configuration requirements: The HTTP Server must be registered with the Oracle Single Sign-On server and the mod_osso module enabled. The registration process typically involves running the ssoreg.sh script at the OSSO server installation and copying the output osso.conf file to the HTTP Server. This process is documented in the Oracle Single Sign-On administration documentation.

JnlpLaunch requirements: The JnlpLaunch Servlet uses the configuration file, JnlpLaunch.properties, to control its behavior. Due to security considerations, this file must not be published or readable to the general public.

JnlpLaunch.properties has the following configuration entries that apply to Single Sign-On:

- *secret.key* is used to create the temporary password, this property should contain a random string. If JnlpLaunch is deployed in a different JVM than the RPM Server EJBs, this string must be an exact match between the JnlpLaunch Servlet and the one available to the RPM EJBs. For security purposes, each separate instance of the RPM application (for example, test versus development) should have a different secret key.
- *user.validation.timeout* indicates the number of seconds the RPM Server uses to determine if a temporary password is still valid.

The JnlpLaunch.properties file is initialized by the RPM installer and should contain valid entries for SSO when the "Enable Single Sign-On in RPM?" prompt was answered by a Y or Yes. However, an administrator may want to alter the `user.validation.timeout` or other property after the initial installation.

When the Oracle Retail RPM installation has finished, go to the WebLogic Administration Console and make sure that the RPM JDBC Datasources and RPM JMS Servers are up and running. On the Deployments Screen, RPM deployment should be active.

To get RPM working with Single Sign On functionality, RPM needs to get protected. WebLogic Tier provides the functionality needed to protect RPM. There are two files in WebLogic Tier that need to be adjusted: mod_wl_ohs.conf and mod_osso.conf . These files are located here:

<ORACLE_INSTANCE>/ config/OHS/ohs1/moduleconf and
<ORACLE_INSTANCE>/ config/OHS/ohs1
Where <ORACLE_INSTANCE> is the instance that is created during installation of Oracle WebTier.
The entries for mod_wl_ohs.conf should like this:

<Location /rpm-client >

SetHandler weblogic-handler

</Location>

The entries for mod_osso.conf should like this:

<Location /rpm-client/client >

WebLogicHost hostname.com

WebLogicPort managed server port number

</Location>

<Location /rpm-client/launch >

WebLogicHost hostname.com

WebLogicPort managed server port number

require valid-user

AuthType Osso

</Location>

Finally, the OHS in Web Tier must be restarted. Go to <ORACLE_INSTANCE>/bin and start the OHS server with: opmnctl startproc ias-component=ohs1. The URL for SSO RPM would be something like this: http://hostname.com:OHSportnumber/rpm-client/launch?template=rpm_jnlp_template.vm

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 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/<evw89t>/war/client/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/evw89t/war/client/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.

After signing rpm_client_config.jar, restart the RPM managed server in WebLogic.

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 11g Administration Console.
2. Click **Lock and Edit.**
3. Under Services, click **JTA**.
4. Click the Configuration tab.
5. Under JTA, 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 either XML or 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://redevlv0072:17011/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**.

To add a rpm_system_options_def row required by RPM needs for system use, to the following.

1. Select System Options Default.
2. In the lower right part of the screen, click **Save**.

RPM also includes a status page application that can be used to verify the installation. For details see the "Price Management Status Page" section in the *Oracle Retail Price Management Operations Guide.*

Oracle Retail provides test cases to "smoke test" the installation. See the My Oracle Support document, *Oracle Retail Merchandising Installation Test Cases* (ID 1277131.1).

RPM Batch Scripts

The RPM application installer configures and installs the batch scripts under $WEBLOGIC_DOMAIN_HOME/retail/<rpmdir>/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) 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
promotionPriceChangePublishExport.sh
purgePayloadsBatch.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/11.2.0.2

#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.p12, 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
/retail/rpm13/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 plsql 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.

Upgrade RPM 13.2.x Future Retail Data

Use the following guidelines to determine if you need to upgrade RPM 13.2.x Future Retail data:

- If this is a new installation of RPM and not an upgrade, this section can be skipped.
- If upgrading the RPM application from version 13.2.5 and the GenerateFutureRetailRollUpBatch.sh data conversion batch process has been successfully run previously or RPM 13.2.6 was a new installation of RPM and not an upgrade from a previous version, this section can be skipped.
- If upgrading the RPM application from version 13.2.3.1 or a previous version, the steps detailed below need to be followed.

> **Note:** If you are running the Future Retail Data Upgrade for 13.2.6, you must unzip the rpm13dataconversion.zip file and apply the 15848953 patch BEFORE running the conversion scripts.

RPM 13.2.x Future Retail Data Upgrade Steps

Prior to upgrading the Future Retail data, all RMS and RPM installation steps for this release are required to be completed successfully.
Once all RMS and RPM application code and database changes have been completed, the following manual steps need to be completed:

1. Extract the contents of the "rpm13dataconversion.zip" archive that is packaged with RPM 13.
2. Verify that the RPM_ITEM_LOC table contains records for all corresponding records on RMS' ITEM_LOC table where the location is a stockholding location and the items are transaction level, approved and sellable. If any data is missing from the RPM_ITEM_LOC table, this data needs to be created on the table prior to executing the next steps.
3. From a SQL*Plus command prompt, execute the RPM_data_conversion_pre_batch.sql script. This script is packaged in the "rpm13dataconversion.zip" archive.
4. The batch process that will convert the Future Retail data is a Java batch process and is threaded using an existing record on the RPM_BATCH_CONTROL table. Verify that the entry for "com.retek.rpm.app.bulkcc.service.BulkConflictCheckAppService" on this table has a value for the THREAD_LUW_COUNT field. If there is no value specified for this field, update the record to have a value that is appropriate for the installation. This should take into consideration hardware, networking, and so on.
5. The batch process that will convert the Future Retail data is multi-threaded and uses a different record on the RPM_BATCH_CONTROL table to determine how many concurrent threads to run. The record for "com.retek.rpm.batch.GenerateFutureRetailRollupBatch" should be updated so that the NUM_THREADS field has a value equivalent to the number of threads used by the conflict checking engine. This can be found in the TaskMDB settings.
6. Execute the GenerateFutureRetailRollUpBatch.sh batch process by providing an input parameter for a valid userid and password. No other parameters should be provided when converting all Future Retail data.

> **Note:** The RPM application server must be running in order to execute this batch process.

> **Note:** During the execution of the GenerateFutureRetailRollUpBatch.sh batch process, no other processes should be running within the RPM database – this includes other batch processes and users interacting with the system.

7. Upon successful completion of the GenerateFutureRetailRollUpBatch.sh batch process, execute the RPM_data_conversion_post_batch.sql script from a SQL*Plus command prompt. This script is packaged in the "rpm13dataconversion.zip archive.
8. If it is not desired to keep copies of the original Future Retail tables after successfully converting data, the following three tables can be dropped from the schema:
   - RPM_FUTURE_RETAIL_ORG
   - RPM_CUST_SEGMENT_PROMO_FR_ORG
   - RPM_PROMO_ITEM_LOC_EXPL_ORG

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;
```

For XML authentication, complete the following steps.

1. Insert entry into users_rsm.xml file.

   <user firstname="firstn" lastname="lastn" username="newuser1"/>

2. Insert entry into ORACLE java wallet. For example,

```
./save_credential.sh -l
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/APPDomain/retail/rpm13/
config -a NEWUSER1 -u newuser1 -p rpm13
```

> Note: The alias after -a must be completely capitalized (for example, NEWUSER1).

3. 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,
       'newuser1',
       -1001,
       nvl(get_vdate,sysdate) - 365,
       null
  from dual;
```

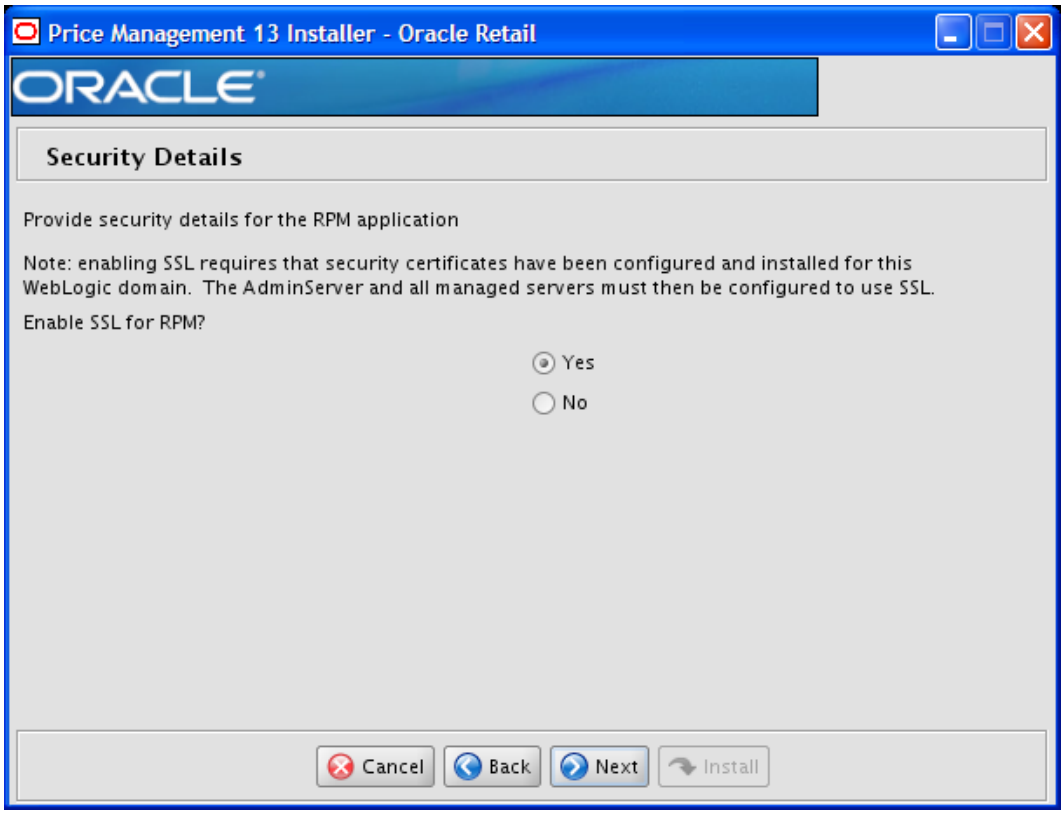
Note: If you are using save_credential.sh to add a wallet entry or to update a wallet entry (or you are adding a user to users_rsm.xml), bounce the application/managed server so your changes are visible to the application. Also, save a backup copy of your cwallet.sso file and users_rsm.xml in a location outside of the deployment path, because a redeployment or reinstallation of the application will wipe out the wallet entries you made after installation of the application. To restore your wallet entries after redeployment or reinstallation, copy the backed up cwallet.sso file over the cwallet.sso file, and copy your backed up users_rsm.xml over the users_rsm.xml. Then bounce the application/managed server.

A

Appendix: RPM Application Installer Screens

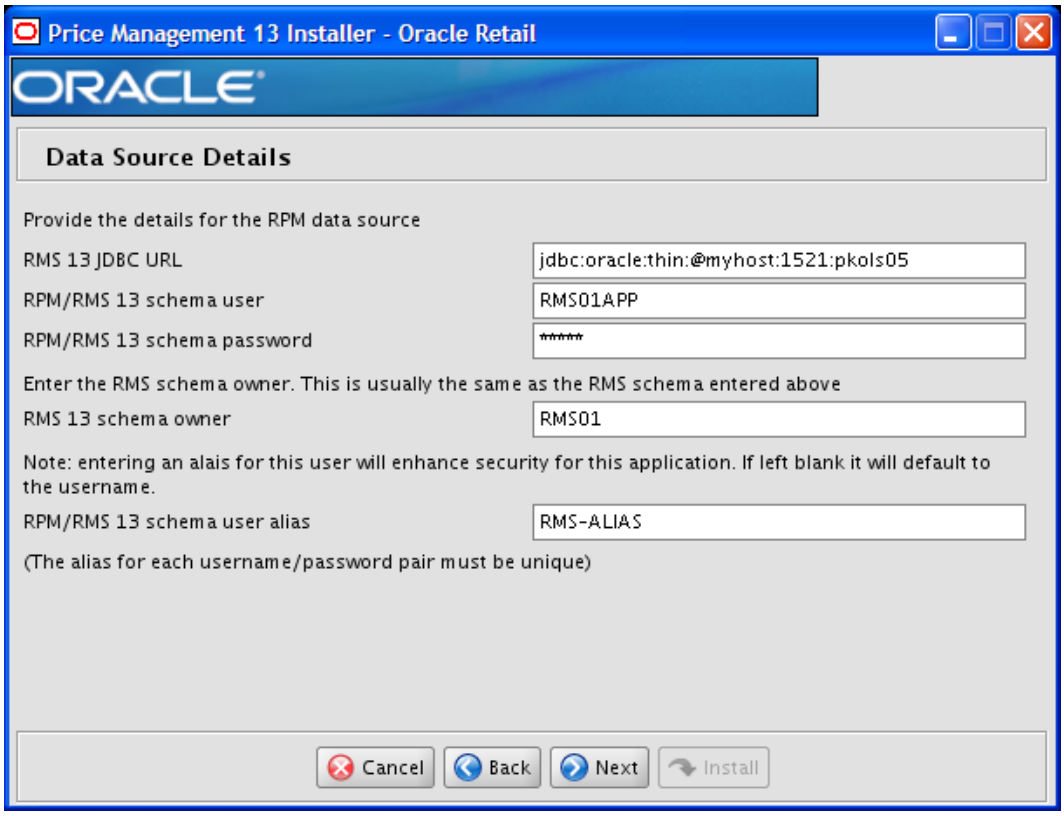
You need the following details about your environment for the installer to successfully deploy the RPM application. Depending on the options you select, you may not see some screens or fields.

Screen: Security Details



Field Title	Enable SSL for RPM?
Field Description	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. 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.

Screen: Data Source Details



Field Title	RMS 13 JDBC URL
Field Description	URL used by the RPM application to access the RMS database schema. See Appendix: URL Reference for expected syntax. **Note:** The RPM database tables are a part of the RMS schema.
Destination	data-sources.xml
Examples	jdbc:oracle:thin:@myhost:1521:pkols05

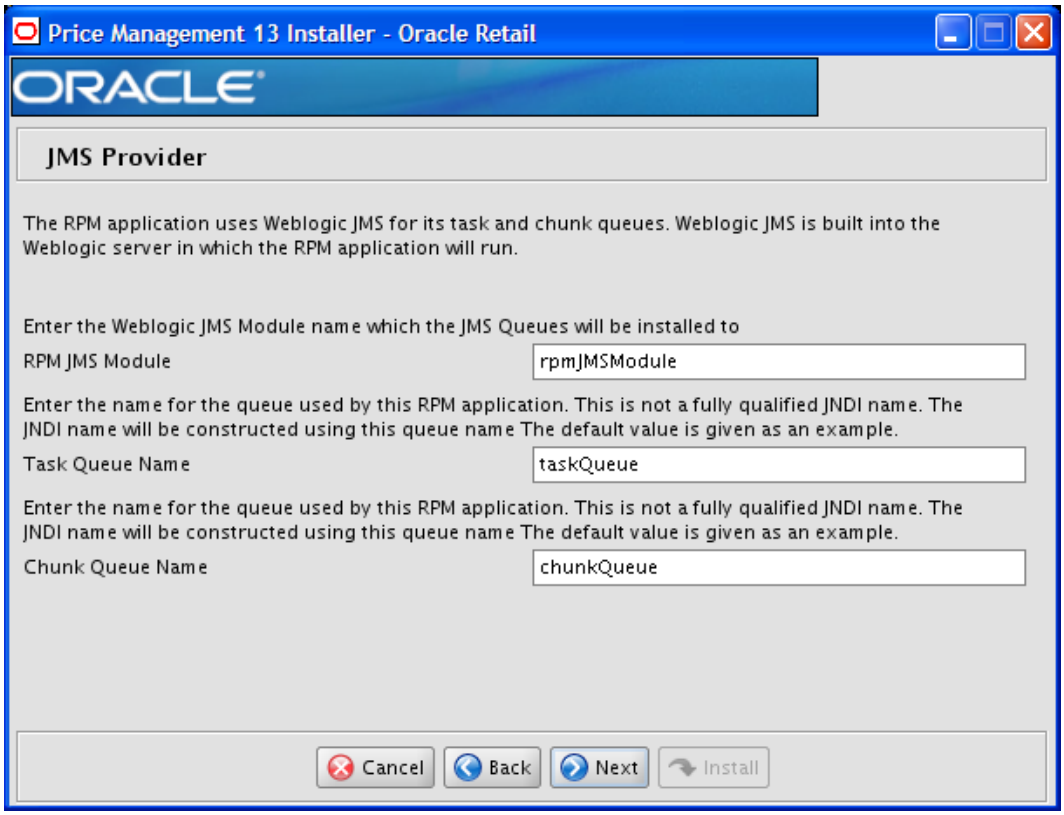
Field Title	RPM/RMS 13 schema user
Field Description	Database user where the RMS database schema was installed.
Destination	data-sources.xml and ORACLE java wallet file
Example	RMS01APP

Field Title	RPM/RMS 13 schema password
Field Description	Password for the RMS schema user.
Destination	ORACLE java wallet file

Field Title	RMS 13 schema owner
Field Description	Database user which owns the RMS tables. This is usually the same as the RMS 13 schema above.
Destination	rpm.properties
Example	RMS01

Field Title	RPM/RMS 13 schema alias
Field Description	Database user which owns the RMS tables. This is usually the same as the RMS 13 schema above.
Destination	rpm.properties and ORACLE java wallet file
Example	RMS-ALIAS
Notes	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: JMS Provider

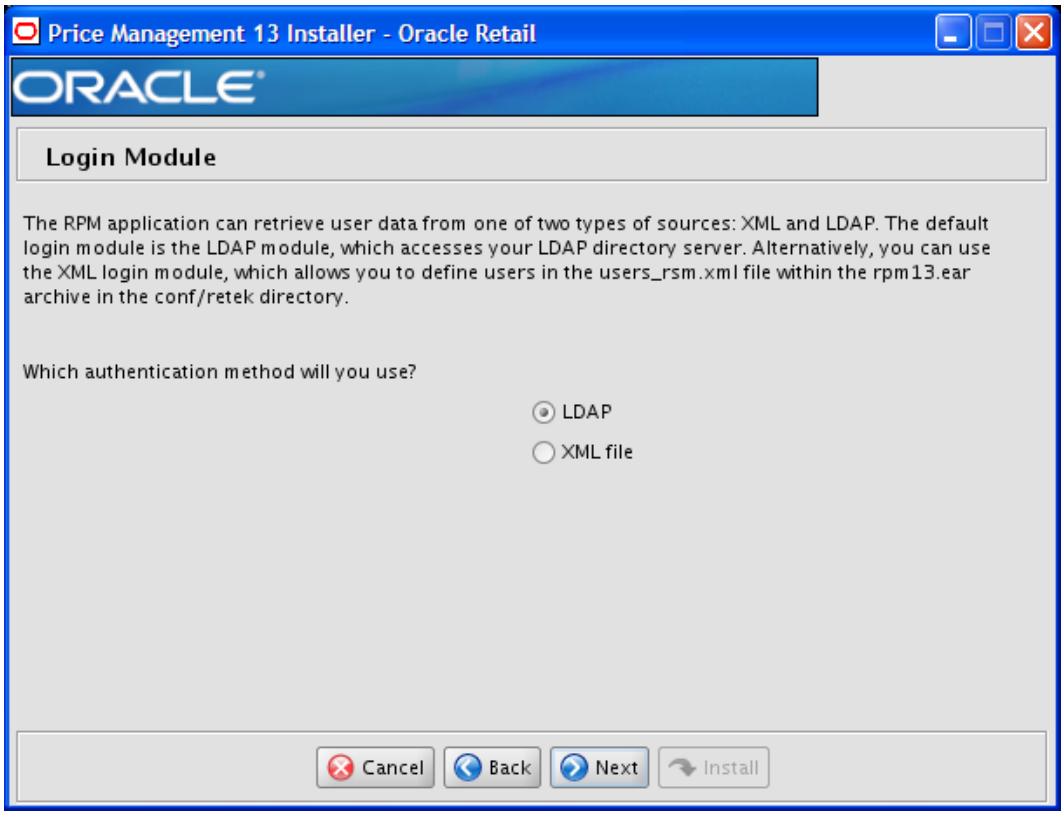


Field Title	RPM JMS Module
Field Description	The WebLogic JMS Module name to where the JMS Queues will be installed.
Destination	rpm.properties and Weblogic server Administration Console.
Example	rpmJMSModule

Field Title	Task Queue Name
Field Description	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). **Note:** This is not a complete JNDI name. The value provided will be appended to jms/ to form the full JNDI name for the queue in the OC4J JMS server.
Destination	rpm.properties and Weblogic server Administration Console.
Example	taskQueue

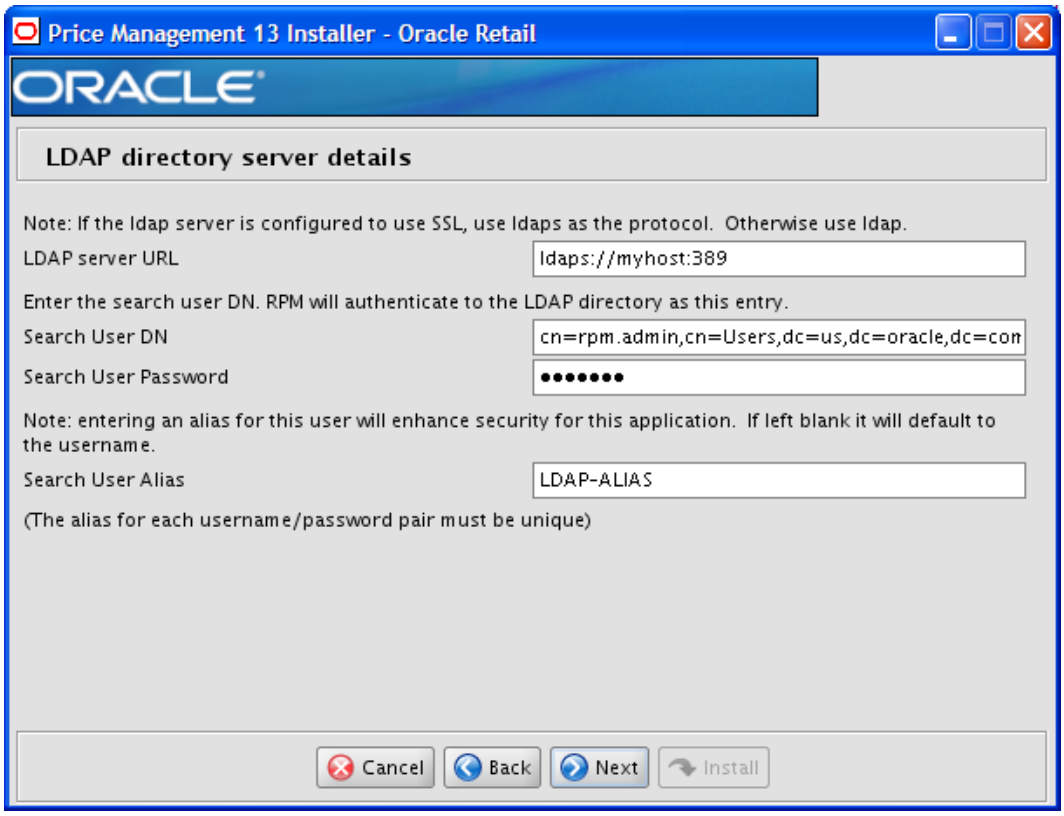
Field Title	Chunk Queue Name
Field Description	Name by which the task queue will be identified. If this is a new RPM environment, choose a queue name that is not already in use in the JMS server. If you have already created the queue in the JMS server as part of the Clustering Preinstallation steps, you must provide the same name in this field (without the jms/ prefix). **Note:** This is not a complete JNDI name. The value provided will be appended to jms/ to form the full JNDI name for the queue in the OC4J JMS server.
Destination	rpm.properties and Weblogic server Administration Console.
Example	chunkQueue

Screen: Login Module



Field Title	Which authentication method will you use?
Field Description	Choose whether the RPM application will authenticate users against an LDAP directory or an XML file on the server.
Destination	security.properties, dao_rpm.xml
Example	LDAP

Screen: LDAP directory server details



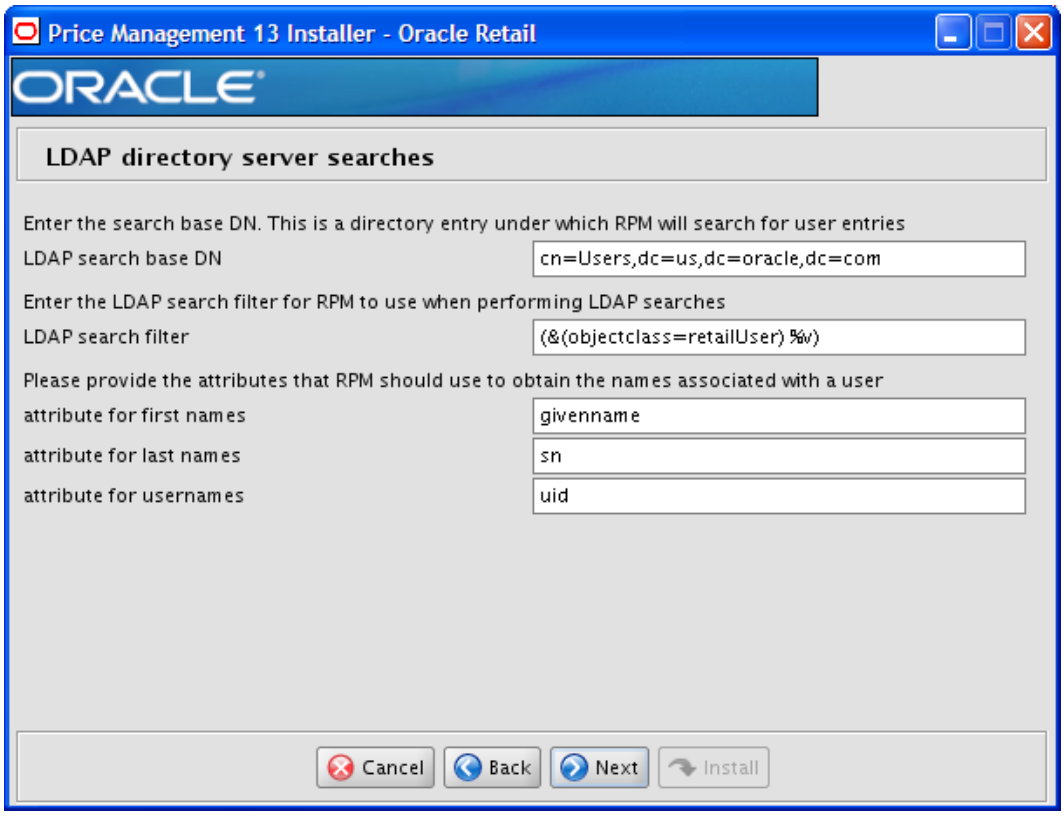
Field Title	LDAP server URL
Field Description	URL for your LDAP directory server. See Appendix: URL Reference for expected syntax.
Destination	security.properties
Example	ldaps://myhost:389/

Field Title	Search User DN
Field Description	Distinguished name of the user that RPM uses to authenticate to the LDAP directory.
Destination	security.properties
Example	cn=rpm.admin,cn=Users,dc=us,dc=oracle,dc=com

Field Title	Search User Password
Field Description	Password for the search user DN.
Destination	security.properties

Field Title	Search User Alias
Field Description	The alias for the search user DN.
Destination	security.properties
Example	LDAP-ALIAS
Notes	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



Field Title	LDAP search base DN
Field Description	Distinguished name of the LDAP directory entry under which RPM should search for users.
Destination	security.properties
Example	cn=Users,dc=us,dc=oracle,dc=com

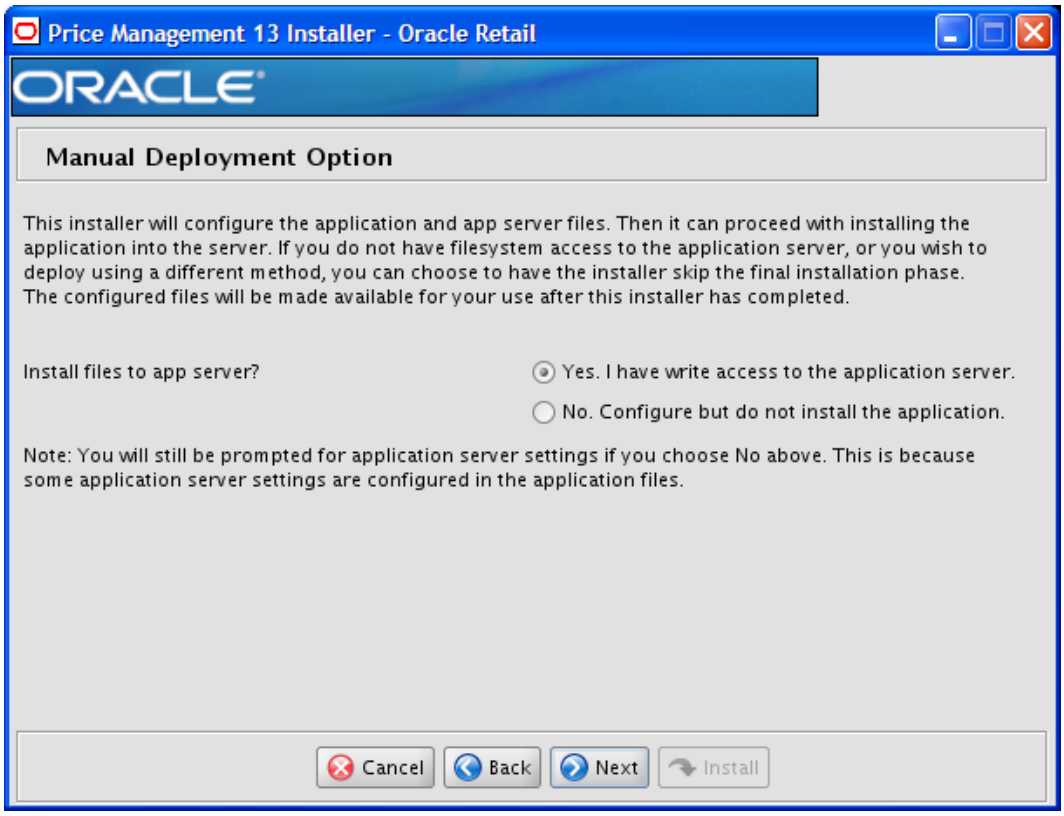
Field Title	LDAP search filter
Field Description	LDAP filter that determines which entries are returned to RPM when it conducts a directory search under the search base DN. See the *Oracle Retail Price Management Operations Guide* for additional information on configuring this field.
Destination	security.properties
Example	(&(objectclass=retailUser) %v)

Field Title	attribute for first names
Field Description	LDAP attribute where RPM should look for a user's first name
Destination	security.properties
Example	givenname

Field Title	attribute for last names
Field Description	LDAP attribute where RPM should look for a user's last name
Destination	security.properties
Example	sn

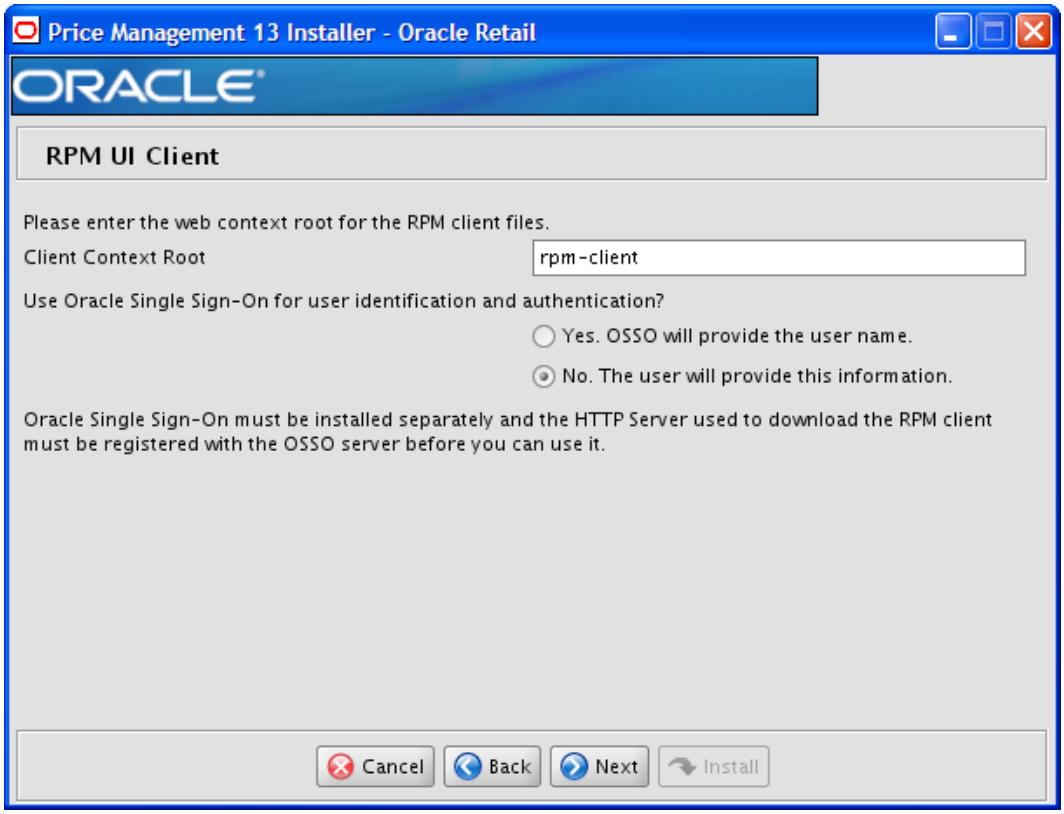
Field Title	attribute for usernames
Field Description	LDAP attribute where RPM should look for a user's username
Destination	security.properties
Example	uid

Screen: Manual Deployment Option



Field Title	Install files to app server?
Field Description	If you do not have write access under ORACLE_HOME, you can still use the installer to gather your settings and configure the RPM files locally in the staging area. Then, at a later time, an administrator can manually copy over the RPM files and deploy the ear file. If you select this option, instructions are printed to the console and the installer log file for the steps needed to complete the installation.
Note	Select Yes. There is a known issue when selecting No. If you choose the option, **No. Configure but do not install the application**, in the installer screen named **Manual Deployment Option**, files required for manual application installation are deleted at the end of the installation.

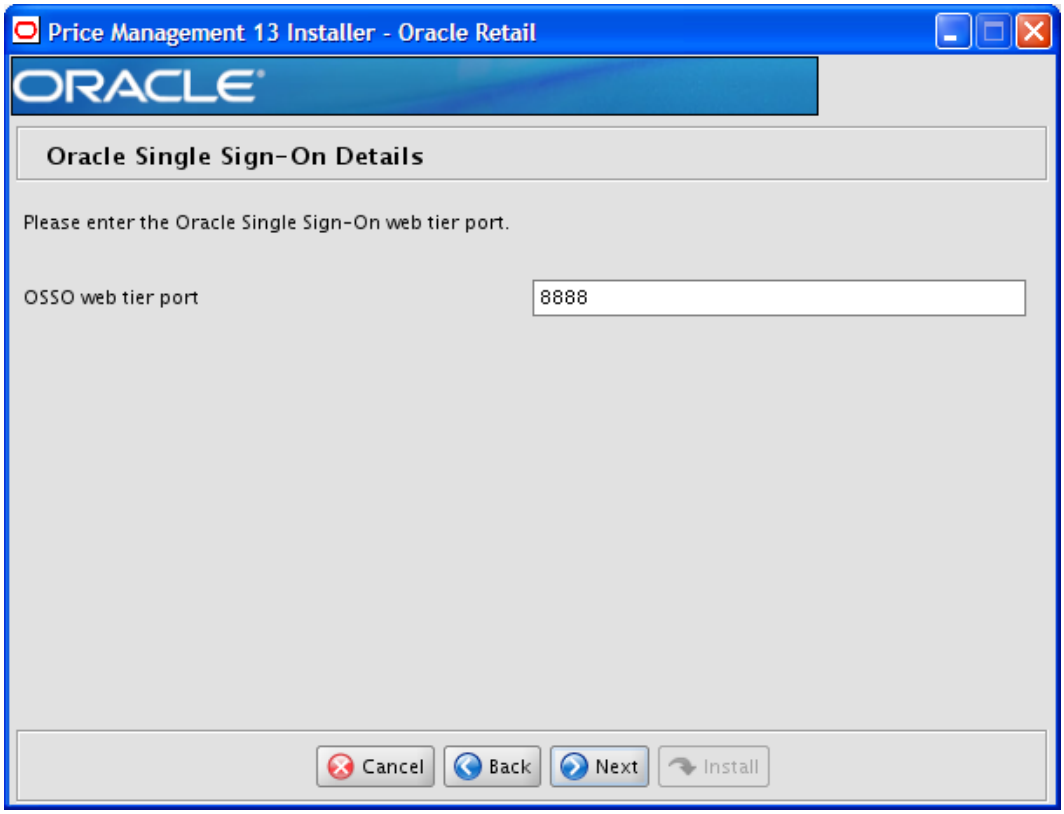
Screen: RPM UI Client



Field Title	Client Context Root
Field Description	The Client Context Root determines how the RPM client will be accessed from users' web browsers. The RPM client URL has the following format: http:/<host>:<port>/<rpm_client_ctx_root>/launch?template=rpm_jnlp_template.vm Example, with RPM Client Context Root value of rpm-client: http://redevlv0072:17011/rpm-client/launch?template=rpm_jnlp_template.vm
Example	rpm-client

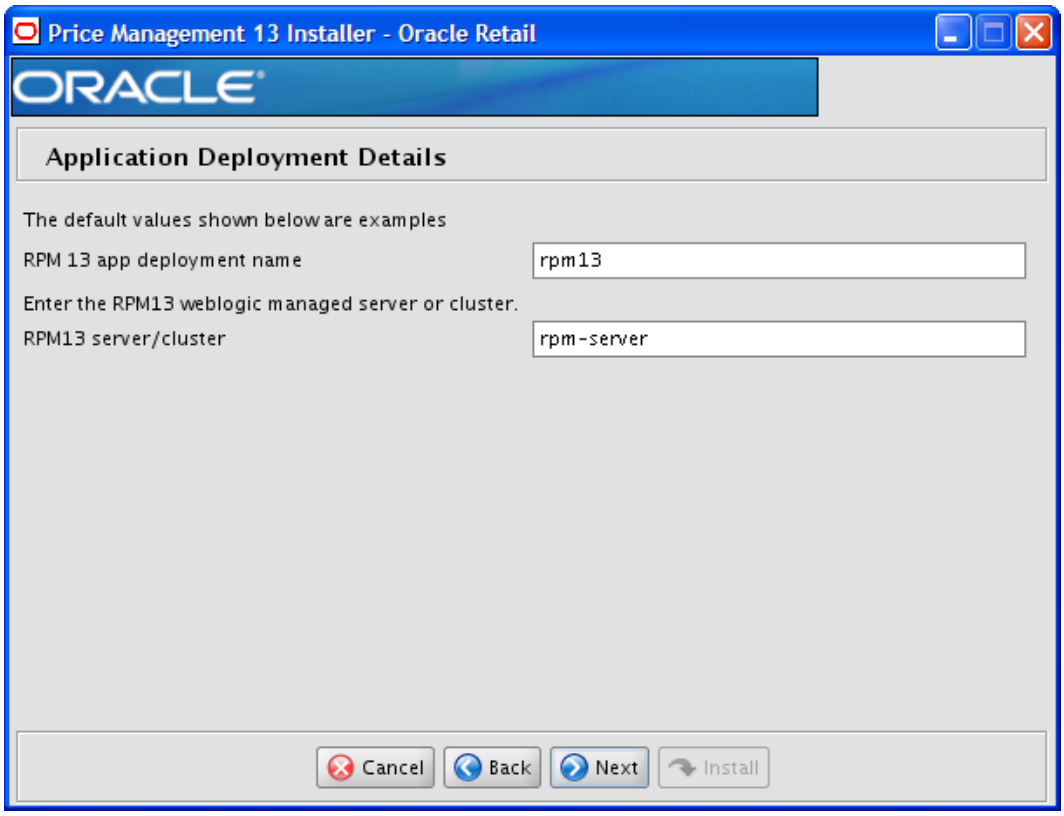
Field Title	Use Oracle Single Sign-On for user identification and authentication?
Field Description	This version of RPM has the option to use Oracle Single Sign-On (OSSO) technology to authenticate users. If OSSO is being used in your environment, choose Yes. The No option configures RPM to use its own LDAP directory settings for authentication.
Destination	JnlpLaunch.properties
Example	No

Screen: Oracle Single Sign-On Details



Field Title	OSSO web tier port
Field Description	Port name for OSSO Web Tier.
Example	8888

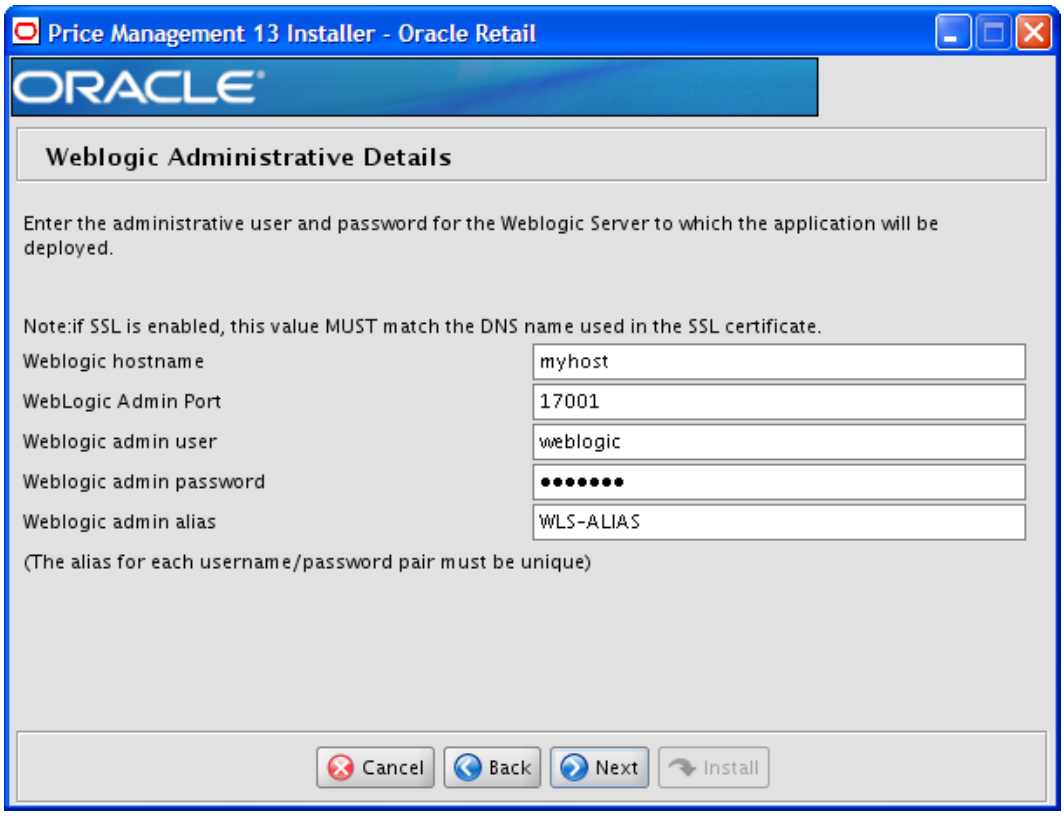
Screen: Application Deployment Details



Field Title	RPM 13 app deployment name
Field Description	Name by which this RPM application is identified in the application server. This value must match the application deployment name/context root name used to update the weblogic.policy file described in the "Install NodeManager" section of this guide. If these values do not match, the application will not run after installation.
Example	rpm13

Field Title	RPM 13 server/cluster
Field Description	Name of the server/cluster that was created for this RPM application. The deployment name given for the RPM 13 app deployment name field should be a member of this server or cluster. 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.
Example	rpm-server

Screen: WebLogic Administrative Details



Field Title	Hostname
Field Description	Hostname of the application server. If SSL is used, this must match the DNS name in the SSL certificate.
Example	myhost

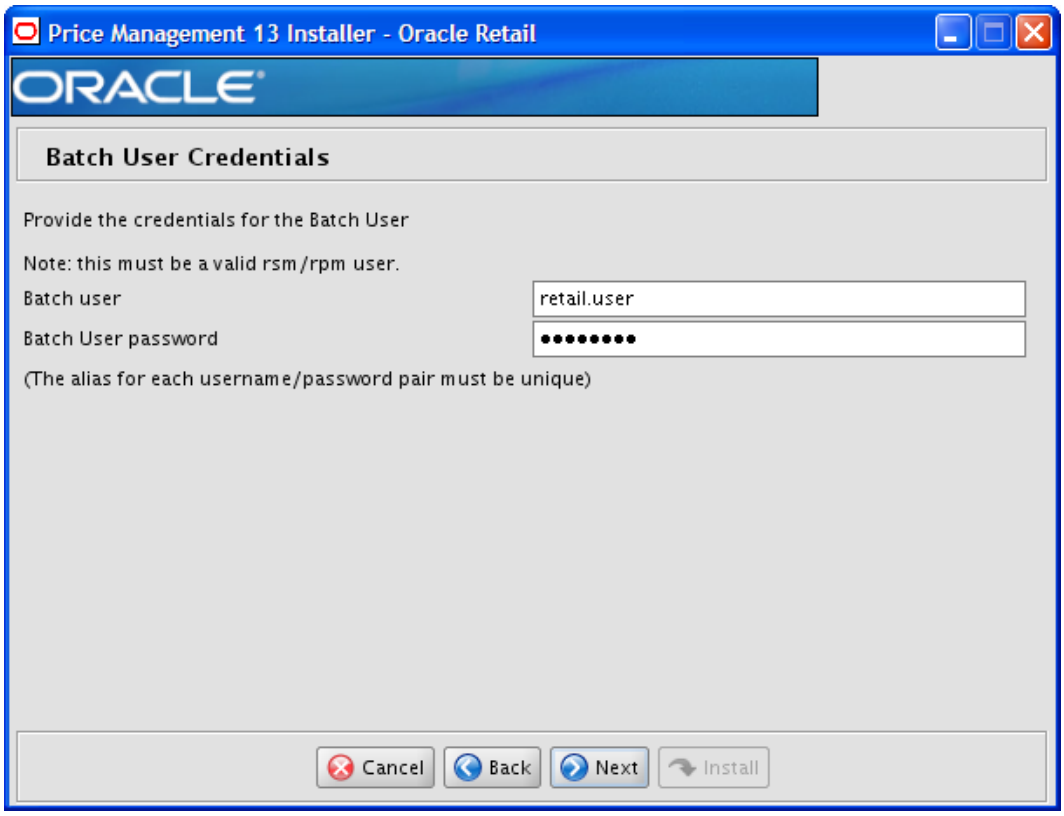
Field Title	WebLogic admin port
Field Description	Listen port for the WebLogic Admin server
Example	17001

Field Title	WebLogic admin user
Field Description	Username of the admin user for the WebLogic instance to which the ReIM application is being deployed.
Example	weblogic

Field Title	WebLogic admin password
Field Description	Password for the WebLogic admin user. You chose this password when you created the WebLogic instance or when you started the instance for the first time.

Field Title	WebLogic admin alias
Field Description	An alias for the WebLogic admin user that is used for ORACLE java wallet.
Example	WLS-ALIAS
Notes	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.

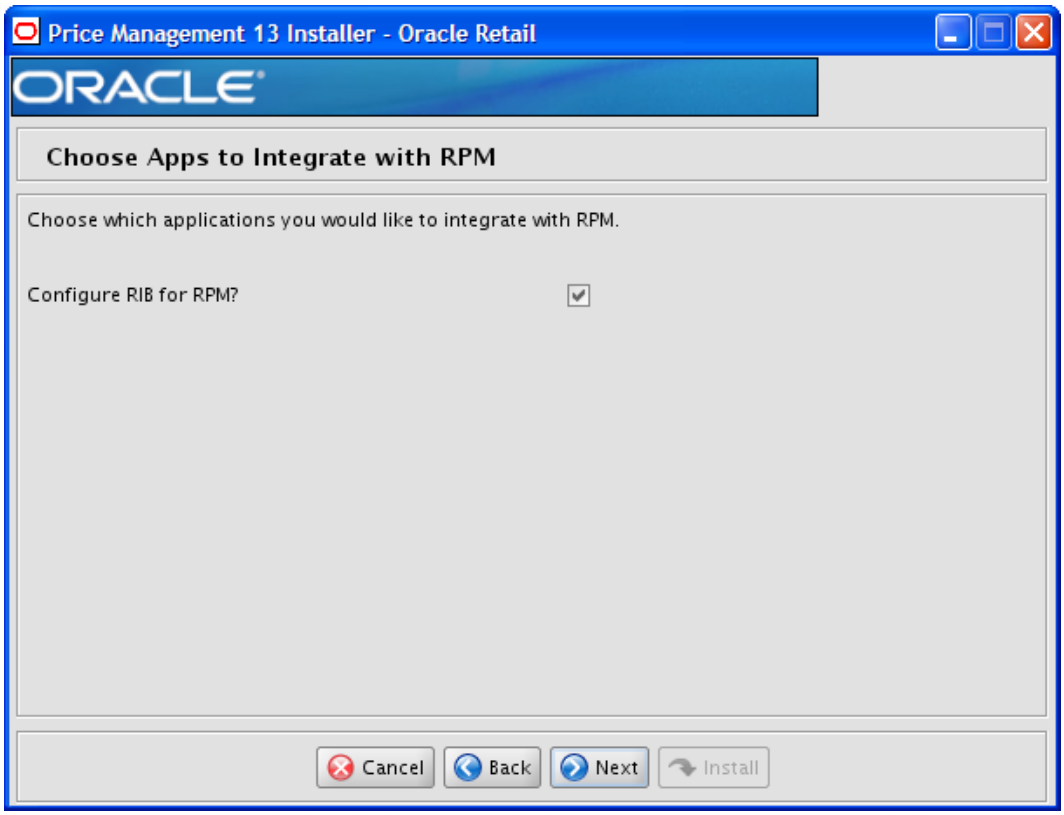
Screen: Batch User Credentials



Field Title	Batch User
Field Description	The RPM user name of the person running RPM batch. It must be a valid RPM user that already exists in the database, or will be coming through LDAP, or will be built in XML authentication. It does not have to exist already in the database, but it must exist when you try to use the alias created in this step to run batch. Using one of the user names you will supply in subsequent screens (such as Setup Application Users) is recommended.
Example	retail.user

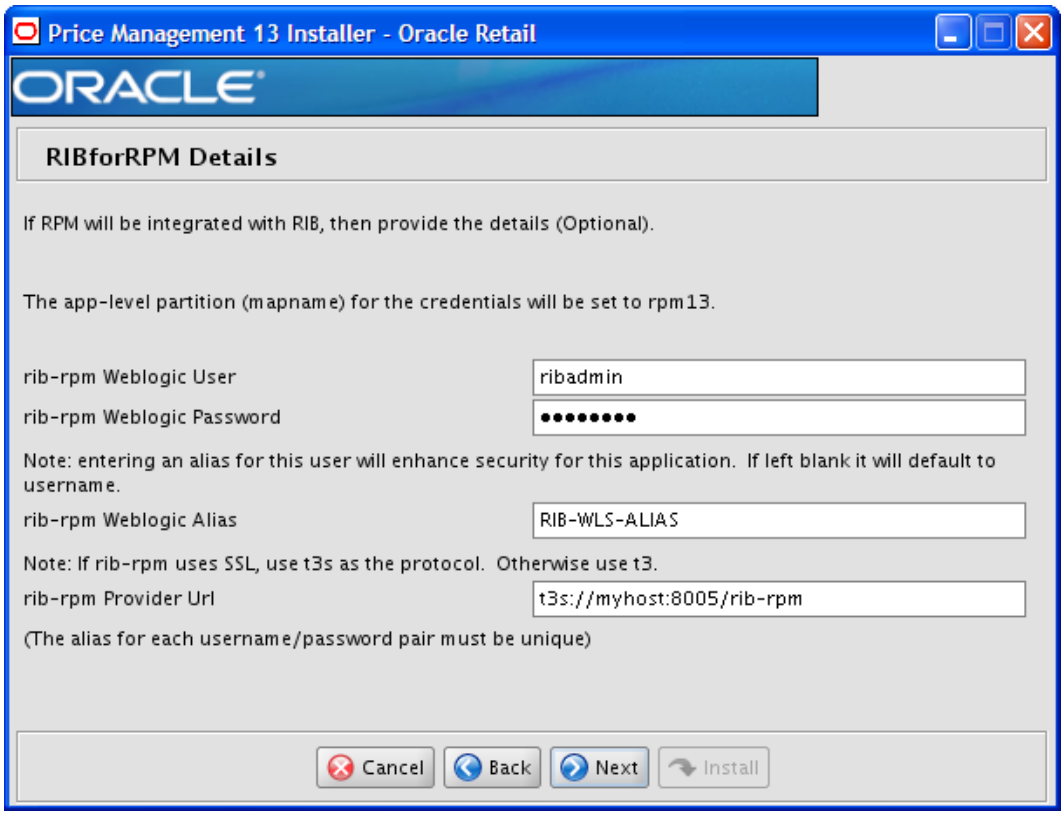
Field Title	Batch User Password
Field Description	The password of the batch user.

Screen: Choose Apps to Integrate with RPM



Field Title	Configure RIB for RPM?
Field Description	Select this option if you will be using RIB with RPM.

Screen: RIBforRPM Details



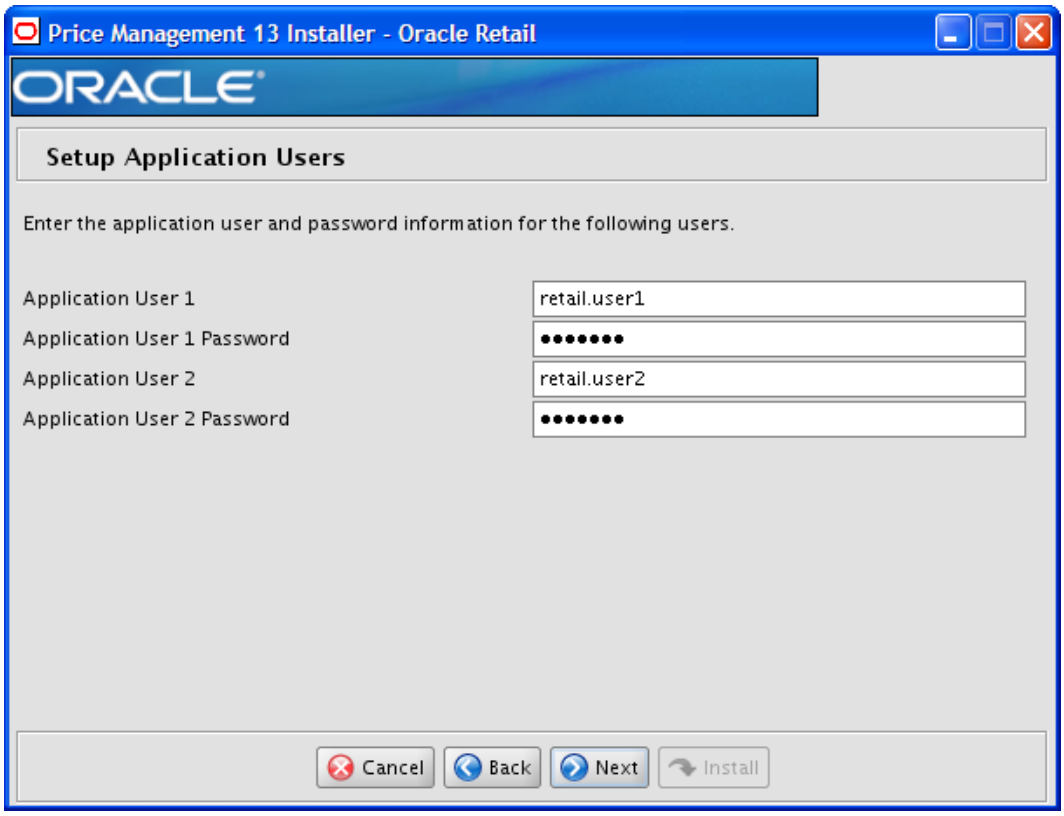
Field Title	rib-rpm WebLogic User
Field Description	The username of the rib-rpm WebLogic user.
Example	ribadmin

Field Title	rib-rpm WebLogic password
Field Description	Password for the RIBforRPM 13 user.

Field Title	rib-rpm WebLogic Alias
Field Description	The alias for the rib-rpm WebLogic user.
Example	RIB-WLS-ALIAS
Notes	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.

Field Title	rib-rpm Provider URL
Field Description	RPM provider URL for rib-rpm
Examples	t3s://myhost:8005/rib-rpm

Screen: Setup Application Users



Field Title	Application User 1

Field Description	An RPM application user name. For XML authentication, the Application User entered is stored in the users_rsm.xml file. This information also is stored with the alias and the password in the ORACLE java wallet. For either XML or LDAP authentication, a row must be built in the database table rsm_user_role in order to work in RSM/RPM. The following is an example of how to build that row.
	`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;`

Example	retail.user1

Field Title	Application User 1 Password
Field Description	The password for the RPM application user.

Field Title	Application User 2

Field Description	An RPM application user name. For XML authentication, the Application User entered is stored in the users_rsm.xml file. This information also is stored with the alias and the password in the ORACLE java wallet. For either XML or LDAP authentication, a row must be built in the database table rsm_user_role in order to work in RSM/RPM. The following is an example of how to build that row.
	`insert into rsm_user_role`
	`(id, user_id, role_id, start_date_time, end_date_time)`
	`select rsm_user_role_seq.nextval,`
	`'retail.user2',`
	`-1001,`
	`nvl(get_vdate,sysdate) - 365,`
	`null`
	`from dual;`

Example	retail.user2

Field Title	Application User 2 Password
Field Description	The password for the RPM application user.

B

Appendix: Installer Silent Mode

In addition to the GUI and text interfaces of the RPM installer, there is a silent mode that can be run. This mode is useful if you wish to run a repeat installation attempt without going through the installer screens again.

The installer runs in two distinct phases. The first phase involves gathering settings from the user. At the end of the first phase, a properties file named ant.install.properties is created with the settings that were provided. Then the second phase begins, where this properties file is used to provide your settings for the installation.

To skip the first phase and re-use the ant.install.properties file from a previous run, follow these instructions:

1. Edit the ant.install.properties file and correct any invalid settings that may have caused the installer to fail in its previous run.
2. Run the installer again with the silent argument.

Example: install.sh silent

C

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 13.2 (See Chapter 2, "RAC and Clustering." Run these scripts in the RSM schema and try logging into RPM again.

Warning: Could not create system preferences directory

Symptom

The following text appears in the installer Errors tab:

```
May 22, 2010 11:16:39 AM java.util.prefs.FileSystemPreferences$3 run
WARNING: Could not create system preferences directory. System preferences are
unusable.
May 22, 2010 11:17:09 AM java.util.prefs.FileSystemPreferences
checkLockFile0ErrorCode
WARNING: Could not lock System prefs. Unix error code -264946424.
```

Solution

This is related to Java bug 4838770. The /etc/.java/.systemPrefs directory may not have been created on your system. See http://bugs.sun.com for details.

This is an issue with your installation of Java and does not affect the Oracle Retail product installation.

ConcurrentModificationException in Installer GUI

Symptom

In GUI mode, the errors tab shows the following error:

```
java.util.ConcurrentModificationException
            at
java.util.AbstractList$Itr.checkForComodification(AbstractList.java:448)
            at java.util.AbstractList$Itr.next(AbstractList.java:419)
… etc
```

Solution

You can ignore this error. It is related to third-party Java Swing code for rendering of the installer GUI and does not affect the retail product installation.

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=<domain_path>/servers/<managed_server>
/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.

RPM displays a red screen with SSO text on top left

Symptom

After you installed RPM and launched it, you get a red screen telling that RPM works only on Single Sign On mode. This error may occur when you are installing RPM for a second time after the first installation failed. What happens is that the first time RPM was installed, it created some directories that were not deleted before starting the second installation.

Solution

To fix the problem you do not have to redeploy RPM. Follow these instructions:

1. Go to your RPM managed server in $WEBLOGIC_DOMAIN_HOME/servers/<rpm-managed-server>/log.
2. Look in the root of /log for the directory velocity.log. If the file name exists as velocity.log, rename that velocity.log file and create a directory with the name velocity.log. If it does not exist create a velocity.log file in that location. It should look like this:

   ```
   WEBLOGIC_DOMAIN_HOME/servers/rpm-server/log/velocity.log
   ```

   Note: velocity.log is a directory.

3. Stop and start the RPM application from the Deployments screen in the WebLogic Server Administration Console and launch RPM again in a new browser.

Installers fail because of missing .jar file in $ORACLE_HOME/utils/ccr/lib

Symptom

The jar file expected by the installer (emocmclnt.jar) is overwritten after the OPatch patch 6880880 is applied, and any other patch is applied afterwards using that OPatch. If you try running the installer after patching as outlined in the installation guides for forms based applications, the installer fails. All applications that are installed in the same WebLogic server that hosts any of the forms applications will be affected by this issue. This is because of the required Oracle patches for Linux 64-bit systems that are applied to the WebLogic forms server using OPatch.

Solution

Back up the content of the $ORACLE_HOME/utils/ccr/lib directory prior to applying OPatch patch 6880880, and recopy the content back after you apply any patches using that opatch.

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.

Clustered installation fails when Node 1 of the cluster is down

Symptom

When RPM is installed as a cluster, the client will fail if the first node of the cluster is down, even if all the rest of the nodes in the cluser are up. This is a known issue with how the installer configures the JNLP files and will be corrected in a future release of RPM.

Solution

This will need to be fixed in the deployed ear file. These instructions assume that the staging mode is set to "stage" and the default application name (rpm13) was used for the deployment. To change the JNLP properties to connect to the RPM server from the node the client was launched from do the following:

The below example will change the deployment on redevlv0124 (node 2) of the cluster, this will also need to be done on redevlv0123 (node 1). If there are more nodes in the cluster this will have to be done to those deployments as well.

1. Stop all the rpm servers
2. Copy the rpm13 directory in the DOMAIN_HOME/servers/rpm-124/stage/rpm13 to a workarea:

   ```
   # cp -r
   /u00/webadmin/product/10.3.3/WLS/user_projects/domains/APPDomain/servers/rpm-
   124/stage/rpm13/rpm13.ear /tmp/rpm
   ```

   > **Note:** It is recommended that you make a copy of this that will stay unchanged in case you need to fall back.

3. Unjar the rpm ear:

   ```
   # cd /tmp/rpm
   # mkdir work
   # cd work
   # jar xf ../rpm13.ear
   ```

4. Update the conf/rpm.properties:

   ```
   # online help URL
   online_rpm_help_url=http://redevlv0124:17011/rpm13-help/help
   ```

5. Update the conf/JnlpLaunch.properties:

   ```
   # Token properties can be used to replace tokens found within a template.
   # For example, the property 'token.somevalue=hello world' will replace the
   # string, '${somevalue}' with the string 'hello world'.
   # There are two reserved token names: 'user' and 'password'.
   token.rpm_provider_url=t3://redevlv0124:17011
   token.rpm_download_url=http://redevlv0124:17011/rpm-client/client
   ```

6. Unjar the JnlpLaunchServlet.war:

   ```
   # cd /tmp/rpm/work
   # mkdir JnlpLaunchServlet.war.work
   # cd JnlpLaunchServlet.war.work
   # jar xf ../JnlpLaunchServlet.war
   ```

7. Update the client/rpmconfig.jnlp

   ```
   <jnlp codebase="http://redevlv0124:17011/rpm-client/client" spec="1.0+"
   href="rpmconfig.jnlp">
   ```

8. Update the template/rpm_jnlp_template.vm

```
    <jnlp codebase="http://redevlv0124:17011/rpm-client/client" spec="1.0+">
    ...
        <property name="client_master.properties"
value="/retek/client_master.properties"/>
        <property name="java.naming.provider.url"
value="t3://redevlv0124:17011"/>
        <property name="NAMING_URL" value="t3://redevlv0124:17011"/>
```

9. Update the rpm_client_config.jar:

```
# cd /tmp/rpm/work/JnlpLaunchServlet.war.work/client/lib
# mkdir rpm_client_config.jar.work
# cd rpm_client_config.jar.work
# jar xf ../rpm_client_config.jar
```

10. Update the rpm.properties

```
# online help URL
online_rpm_help_url=http://redevlv0124:17011/rpm13-help/help
```

11. Create the new jar/war going backwards and sign it:

```
# cd
/tmp/rpm/work/JnlpLaunchServlet.war.work/client/lib/rpm_client_config.jar.work
# jar cf rpm_client_config.jar *
# cd ../
# rm rpm_client_config.jar
# mv rpm_client_config.jar.work/rpm_client_config.jar .
# rm -rf rpm_client_config.jar.work

# jarsigner rpm_client_config.jar foo

# cd /tmp/rpm/work/JnlpLaunchServlet.war.work
# jar cf JnlpLaunchServlet.war *
# cd ..
# rm JnlpLaunchServlet.war
# mv JnlpLaunchServlet.war.work/JnlpLaunchServlet.war .
# rm -rf JnlpLaunchServlet.war.work

# jarsigner JnlpLaunchServlet.war foo

# cd /tmp/rpm/work
# jar cf rpm13.ear *

# jarsigner rpm13.ear foo
```

> **Note:** Be sure to do this for each server on the cluster.

12. Remove the deployment and put the new ear in place to get re-deployed on each server in the cluster.
13. Remove the deployment from the rpm server tmp directory:

```
# cd
/u00/webadmin/product/10.3.3/WLS/user_projects/domains/APPDomain/servers/rpm-
124/tmp/_WL_user
# rm -rf rpm13
```

14. Replace the rpm13.ear from the rpm-124/stage/rpm13 directories with the new ear

```
# cd
/u00/webadmin/product/10.3.3/WLS/user_projects/domains/APPDomain/servers/rpm-
124/stage/rpm13
# rm -f rpm13.ear
# cp /tmp/rpm/work/rpm13.ear .
```

15. After the new rpm13.ear files are all in place on each server in the cluster then start the rpm servers, which will re-deploy the ear file with the new JNLP properties.

D

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/rpm13
>
> **Note:** The JNDI provider URL can have a different format depending on your cluster topology. Consult the WebLogic documentation.

E

Appendix: Setting Up Password Stores with Oracle Wallet

As part of an application installation, administrators must set up password stores for database user accounts using Oracle Wallet. These 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.

A password store for the application and application server user accounts must also be installed; however, the installer takes care of this entire process.

About 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 two different types of password stores or wallets. One type is for database connect strings used in program arguments (such as `sqlplus /@db_username`). The other type is 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, RETL, RMS, RWMS, and ARI
- For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For RMS, RWMS, RPM Batch, 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/rms13.2/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/rms13.2/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/11.2.0.3/network/admin/tnsnames.ora

dvols29_rms01user =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))

dvols29_rms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))
```

> **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/rms13.2/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/rms13.2/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 11

   Connected to:
   Oracle Database 11g

   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 value of the entry:

  ```
  rms01user
  ```

  ```
  mkstore –wrl . –viewEntry oracle.security.client.password1
  ```

  Returns value of the entry:

  ```
  passwd
  ```

For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For Java application, 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: mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/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:

  mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/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:

  mspdv351:[1033_WLS] /u00/webadmin/reim/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:

```
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/java_domain/retail/rpm132test/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.
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/java_domain/retail/rpm132test/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/132_mck_soa_domain/retail/reim13/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/132_mck_soa_domain/retail/reim13/config

```
=========================================

Application level key partition name:reim13
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:rms132mock
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:

```
mspdv351:[1033_WLS]
/u00/webadmin/mock132_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 = reim13 to retrieve the password for application use.

Reim.properties code sample:

```
datasource.url=jdbc:oracle:thin:@mspdv349.us.oracle.com:1521:pkols07
datasource.schema.owner=rms132mock
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/132_mck_soa
_domain/retail/reim13/config
csm.wallet.partition.name=reim
```

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 reimbat, 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>

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-13.2.0`
   - `RFX_TMP=/u00/rfx/rfx-13.2.0/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 `$MMHOME/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.

Quick Guide for Retail Wallets

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (MMHOME)>/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile, execution	Installer	n/a	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile	Installer	n/a	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	n/a	<Db_Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rwms schema owner>	Compile forms, execute batch	Installer	n/a	Alias hard-coded by installer
RPM app	DB	<RPM batch install dir>/.wallet	n/a	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	
RWMS auto-login	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	<RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms13inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	n/a	Alias hard-coded by installer

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip13	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip13	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip13	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm13	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rpm13	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			rpm13	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/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 shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
			<installed app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
RSL app	JAVA	<RSL INSTALL DIR>/rsl-rms/security/config							Each alias must be unique
			rsl-rsm	<rsl weblogic user alias>	<rsl weblogic user name>	App use	Installer	weblogic-alias	
			rsl-rsm	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
SIM app	JAVA	<weblogic domain home>/retail/<deployed sim app name>/config							
			rpm	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rms	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			rib-sim	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RETL	DB	<RETL home>/.wallet	n/a	<target application user alias>	<target application db userid>	App use	Manual	<db>_<user>	User may vary depending on RETL flow's target application
RIB	JAVA	<RIBHOME DIR>/deployment-home/conf/security							<app> is one of aip, rfm, rms, rpm, sim, rwms, tafr
JMS			jms<1-5>	<jms user alias> for jms<1-5>	<jms user name> for jms<1-5>	Integration use	Installer	jms-alias	
WebLogic			rib-<app>-app-server-instance	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integration use	Installer	weblogic-alias	
Admin GUI			rib-<app>#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integration use	Installer	admin-gui-alias	
Application			rib-<app>#user-alias	<app weblogic user alias>	<app weblogic user name>	Integration use	Installer	app-user-alias	Valid only for aip, rpm, sim
DB			rib-<app>#app-db-user-alias	<rib-app database schema user alias>	<rib-app database schema user name>	Integration use	Installer	db-user-alias	Valid only for rfm, rms, rwms, tafr
Error Hospital			rib-<app>#hosp-user-alias	<rib-app error hospital database schema user alias>	<rib-app error hospital database schema user name>	Integration use	Installer	hosp-user-alias	

F

Appendix: Oracle 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 currently provides two different implementations: Oracle Single Sign-On (OSSO), and Oracle Access Manager (provides more comprehensive user access capabilities).

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 Oracle Single Sign-On?

The nexus of an Oracle Single Sign-On system is the Oracle Identity Management Infrastructure installation. This consists of 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 HTTP Server 11g Release 1 as a front end to the Oracle WebLogic Server. The Oracle HTTP Server is included in the Oracle Web Tier Utilities 11g Release 1 (11.1.1).
- An Oracle Single Sign-On Plug-in, used to authenticate the user and create the OSSO session cookie. This is available in the Oracle Fusion Middleware 11g Web Tier Utilities (11.1.1.6) package. For Oracle Forms applications like RMS and RWMS, HTTP server will be used.
- The Delegated Administration Services (DAS) application in OID10g and 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 OSSO system and registering HTTP servers.

Additional WebLogic managed servers will be needed to deploy the business applications leveraging the OSSO technology.

Can Oracle Single Sign-On Work with Other SSO Implementations?

Yes, OSSO 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 OSSO environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic OSSO protection typically display a Login link to provide user authentication and gain greater access to the application's resources.

Identity Management Infrastructure for 10g, Oracle Identity Management (OIM) and Oracle Access Manager (OAM) Oracle Access Manager (OAM) for 11g

If using OSSO 10g, The Identity Management Infrastructure is the collection of product and services which provide Oracle Single Sign-on functionality. For OSSO 10g, this includes the Oracle Internet Directory, an Oracle HTTP server, and the Oracle Single Sign-On services. The Oracle Application Server deployed with these components is typically referred as the Infrastructure instance.

If using SSO with OAM11g, Oracle Identity Management (OIM) 11g includes Oracle Internet Directory and ODSM. Oracle Access Manager (OAM) 11g should be used for SSO using osso agent. Oracle Forms 11g contains Oracle HTTP server and other Retail Applications will use WebTier11g for HTTP.

MOD_OSSO

mod_osso is an Apache Web Server module an Oracle HTTP Server uses to function as a partner application within an Oracle Single Sign-On environment. The Oracle HTTP Server is based on the Apache HTTP Server.

MOD_WEBLOGIC

mod_WebLogic operates as a module within the HTTP server that allows requests to be proxied from the Apache HTTP server to the WebLogic server.

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 Single Sign-On.

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. OHS or WebTier uses the MOD_OSSO module to configure this functionality.

All partner applications must be registered with the Oracle Single Sign-On server if using OSSO10g and all partner applications must be registered with Oracle Access Manager (OAM) 11g if using OAM11g for SSO implementation. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Realm

A Realm is a collection users and groups (roles) managed by a single password policy. This policy controls what may be used for authentication (for example, passwords, X.509 certificates, and biometric devices). A Realm also contains an authorization policy used for controlling access to applications or resources used by one or more applications.

A single OID can contain multiple Realms. This feature can consolidate security for retailers with multiple banners or to consolidate security for multiple development and test environments.

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 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.

> **Note:** Dynamically Protected URL and Statically Protected URL are within the context of the Oracle Software Security Assurance (OSSA). The static protection for URLs is a common JEE feature.

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 OSSO user IDs to a database logins on a per-application basis.

How Oracle Single Sign-On Works

Oracle Single Sign-On involves a couple of different components. These are:

- The Oracle Single Sign-On (OSSO) servlet, 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 HTTP Server associated with the Web application, which verifies and controls browser redirection to the OSSO servlet.
- If the Web application implements dynamic protection, then the Web application itself is involved with the OSSO system.

Statically Protected URLs

When an unauthenticated user accesses a statically protected URL, the following occurs:

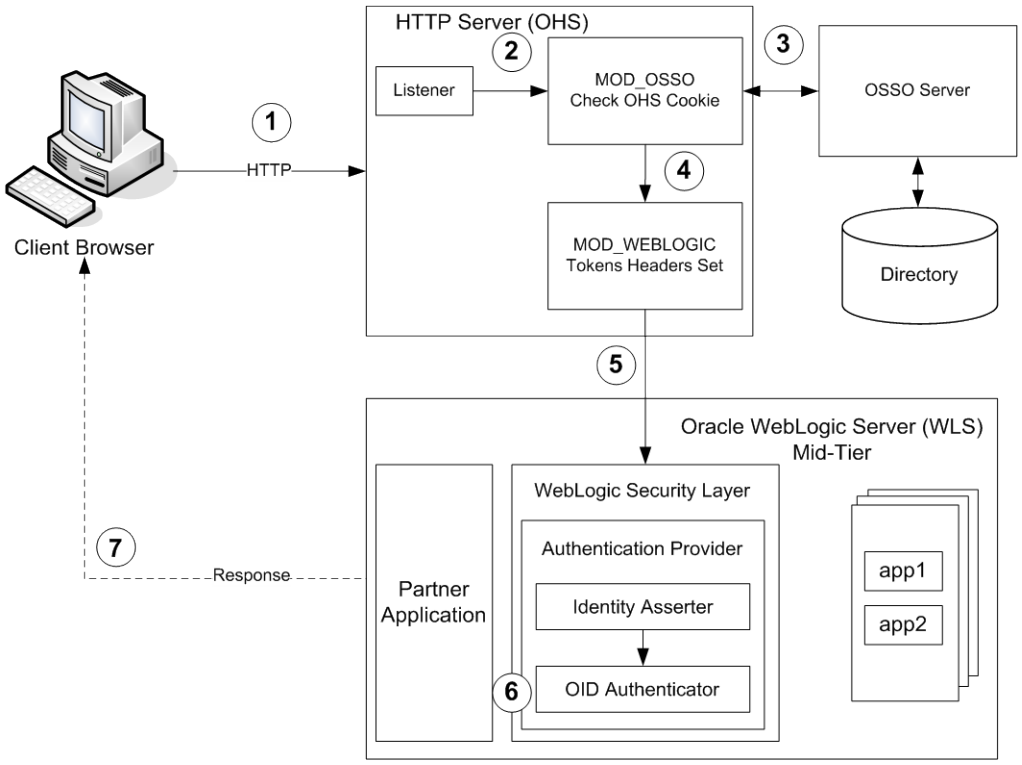
1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS).
2. The Oracle HTTP Server processses the request and routes it to the mod_oss module.
3. This module determines whether the user is already authenticated. If the authentication is required, it directs the browser to the OSSO server. The OSSO server checks for a secure cookie containing the authentication information. If the cookie is not found, the following occurs:
   a. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
   b. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server. The browser is then redirected back to the Oracle HTTP Server with the encrypted authentication credentials. It does NOT contain the user's password.
4. The mod_osso module then decrypts the user credentials and sets HTTP headers with relevant user attributes, marking the user's session as authenticated.
5. The mod_WebLogic module (within the Oracle HTTP Server) then forwards the request to the Oracle WebLogic Server.
6. The Oracle WebLogic Server then invokes the configured authentication providers that decode the headers and provide the user's role membership. In an OSSO implementation, ensure that the OSSO Identity Asserter is invoked and Oracle Internet Directory (OID) Authenticator is executed to provide the user's role membership.
7. Once the authentication is established, the relevant application logic is initiated and the response is sent back to the user through the Oracle HTTP Server.
   Because the Web browser session is now authenticated, subsequent requests in that session are not redirected to the OSSO server for authentication.

Dynamically Protected URLs

When an unauthenticated user accesses a dynamically protected URL, the following occurs:

1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS). The Oracle HTTP server recognizes the user has not been authenticated, but allows the user to access the URL.
2. The application determines the user must be authenticated and send the Oracle HTTP Server a specific status to begin the authentication process.
3. The Oracle HTTP Server processes the request and routes it to the mod_oss module.
4. This module determines whether the user is already authenticated. If the authentication is required, it directs the browser to the OSSO server. The OSSO server checks for a secure cookie containing the authentication information. If the cookie is not found, the following occurs:
   a. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
   b. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server. The browser is then redirected back to the Oracle HTTP Server with the encrypted authentication credentials. It does NOT contain the user's password.
5. The mod_osso module then decrypts the user credentials and sets HTTP headers with relevant user attributes, marking the user's session as authenticated.
6. The mod_WebLogic module (within the Oracle HTTP Server) then forwards the request to the Oracle WebLogic Server.
7. The Oracle WebLogic Server then invokes the configured authentication providers that decode the headers and provide the user's role membership. In an OSSO implementation, ensure that the OSSO Identity Asserter is invoked and Oracle Internet Directory (OID) Authenticator is executed to provide the user's role membership.
8. Once the authentication is established, the relevant application logic is initiated and the response is sent back to the user through the Oracle HTTP Server.
   Because the Web browser session is now authenticated, subsequent requests in that session are not redirected to the OSSO server for authentication.

Single Sign-on Topology



Installation Overview

Installing Oracle Single Sign-On 10g requires installation of the following:

1. Oracle Internet Directory (OID) LDAP server and the Infrastructure Oracle Application Server (OAS). They are typically installed using a single session of the Oracle Universal Installer and are performed at the same time. OID requires an Oracle relational database. If one is not available, the installer will install this as well.

   The Infrastructure OAS includes the Delegated Administration Services (DAS) application as well as the OSSO servlet. The DAS application can be used for user and realm management within OID.
2. Additional midtier instances (such as Oracle Forms 11g) for Oracle Retail applications based on Oracle Forms technologies (such as RMS). These instances must be registered with the Infrastructure OAS installed in step 1. For additional information on SSO 10g installation, see the Creating a High-Availability Environment Whitepaper (My Oracle Support Doc ID: 1311392.1).
3. Additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities.

Installing Oracle Single Sign-On 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 11gR1 (11.1.1.6). The ODSM application can be used for user and realm management within OID.
2. Oracle Access Manager 11gR1 (11.1.1.5) has to be installed and configured.

3. Additional midtier instances (such as Oracle Forms 11g) 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. For additional information on SSO 11g installation, see the Oracle Access Manager and Single Sign-On Whitepaper (My Oracle Support Doc ID 1492047.1).

Infrastructure Installation and Configuration

The Infrastructure installation for OSSO and Oracle Access Manager (OAM) is dependent on the environment and requirements for its use. Deploying an Infrastructure OAS or 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 Application Server Installation Guide and the Oracle Internet Directory Installation Guide (if using OSSO 10g) for more details and Oracle Identity Management Installation Guide11g (if using OAM11).*

OID User Data

Oracle Internet Directory is an LDAP v3 compliant directory server. It provides standards-based user definitions out of the box.

The current version of Oracle Single Sign-On only supports OID as its user storage facility. 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.

OID with Multiple Realms

OID and OSSO can be configured to support multiple user Realms. Each realm is independent from each other and contains its own set of user IDs. As such, creating a new realm is an alternative to installing multiple OID and Infrastructure instances. Hence, a single Infrastructure OAS can be used to support development and test environments by defining one realm for each environment.

Realms may also be used to support multiple groups of external users, such as those from partner companies. For more information on Realms, see the *Oracle Internet Directory Administrators Guide*.

User Management

User Management consists of displaying, creating, updating or removing user information. There are two basic methods of performing user management: LDIF scripts and the Delegate Administration Services (DAS) application available for OID10g or Oracle Directory Services Manager (ODSM) available for OID11g.

OID DAS

The DAS application is a Web-based application used in OID10g is designed for both administrators and users. A user may update their password, change their telephone number of record, or modify other user information. Users may search for other users based on partial strings of the user's name or ID. An administrator may create new users, unlock passwords, or delete users.

The DAS application is fully customizable. Administrators may define what user attributes are required, optional or even prompted for when a new user is created.

Furthermore, the DAS application is secure. Administrators may also what user attributes are displayed to other users. Administration is based on permission grants, so different users may have different capabilities for user management based on their roles within their organization.

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 Single Sign-On 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 Single Sign-On.

G

Appendix: Preinstallation for Secured Setup of RPM in WebLogic

WebLogic Server supports SSL on a dedicated listen port. The managed server can be configured to use SSL as well. To establish an SSL connection, a Web browser connects to WebLogic Server by supplying the SSL listen port and the HTTPs protocol in the connection URL, for example, https://myserver:7002.

RPM deployment is supported in WebLogic in secured mode. For enterprise deployment, it is recommended to use SSL certificates signed by certificate authorities.

> **Note:** Separate signed SSL certificates needs to be obtained for each host where application is being deployed.

Get an SSL Certificate and Set up a Keystore

1. Obtain an identity (private key and digital certificates) and trust (certificates of trusted certificate authorities) for WebLogic Server. Use the digital certificates, private keys, and trusted CA certificates provided by the WebLogic Server kit, the CertGen utility, Sun Microsystem's keytool utility, or a reputable vendor such as Entrust or Verisign to perform this step.
   a. Set appropriate JAVA_HOME and PATH to java.

      Example:

      ```
      export JAVA_HOME=/u00/webadmin/product/jdk
      export PATH=$JAVA_HOME/bin:$PATH
      ```

   b. Create a new keystore.

      ```
      keytool -genkey -keyalg RSA -keysize 2048 -keystore <keystore> -alias
      <alias>
      ```

      Example:

      ```
      keytool -genkey -keyalg RSA -keysize 2048 -keystore redevlv0126.keystore -
      alias redevlv0126
      ```

   c. Generate the signing request.

      ```
      keytool -certreq -keyalg RSA -file <certificate request file> -keystore
      <keystore> -alias <alias>
      ```

      Example:

      ```
      keytool -certreq -keyalg RSA -file redevlv0126.csr -keystore
      redevlv0126.keystore -alias redevlv0126
      ```

   d. Submit the certificate request to Certificate authority
2. Store the identity and trust. Private keys and trusted CA certificates which specify identity and trust are stored in a keystore.

   In following examples, we are using same keystore to store all certificates.

   a. Import the root certificate into the keystore.

      Example:

      ```
      keytool -import -trustcacerts -alias  verisignclass3g3ca -file Primary.pem
      -keystore  redevlv0126.keystore
      ```

b. Import the intermediary certificate (if required) into the keystore.

   Example:

   ```
   keytool -import -trustcacerts -alias oracleclass3g3ca -file Secondary.pem
   -keystore  redevlv0126.keystore
   ```

c. Import the received signed certificate for this request into the keystore.

   Example:

   ```
   keytool -import -trustcacerts -alias redevlv0126 -file cert.cer -keystore
   redevlv0126.keystore
   ```

Configure the Application Server for SSL

1. Configure the identity and trust keystores for WebLogic Server in the WebLogic Server Administration Console.
   a. In the Change Center of the Administration Console, click Lock & Edit.
   b. In the left pane of the Console, expand Environment and select Servers.
   c. Click the name of the server for which you want to configure the identity and trust keystores.
   d. Select Configuration > Keystores.
   e. In the Keystores field, select the method for storing and managing private keys/digital certificate pairs and trusted CA certificates. These options are available:
      – **Demo Identity and Demo Trust**: The demonstration identity and trust keystores, located in the BEA_HOME\server\lib directory and the JDK cacerts keystore, are configured by default. Use for development only.
      – **Custom Identity and Java Standard Trust**: A keystore you create and the trusted CAs defined in the cacerts file in the JAVA_HOME\jre\lib\security directory.
      – **Custom Identity and Custom Trust [Recommended]**: Identity and trust keystores you create.
      – **Custom Identity and Command Line Trust**: An identity keystore you create and command-line arguments that specify the location of the trust keystore.

      Select **Custom Identity and Custom Trust**.
   f. In the Identity section, define attributes for the identity keystore.
      – **Custom Identity Keystore**: The fully qualified path to the identity keystore.
      – **Custom Identity Keystore Type**: The type of the keystore. Generally, this attribute is Java KeyStore (JKS); if left blank, it defaults to JKS.
      – **Custom Identity Keystore Passphrase**: The password you will enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore so whether or not you define this property depends on the requirements of the keystore.
   g. In the **Trust** section, define properties for the trust keystore.

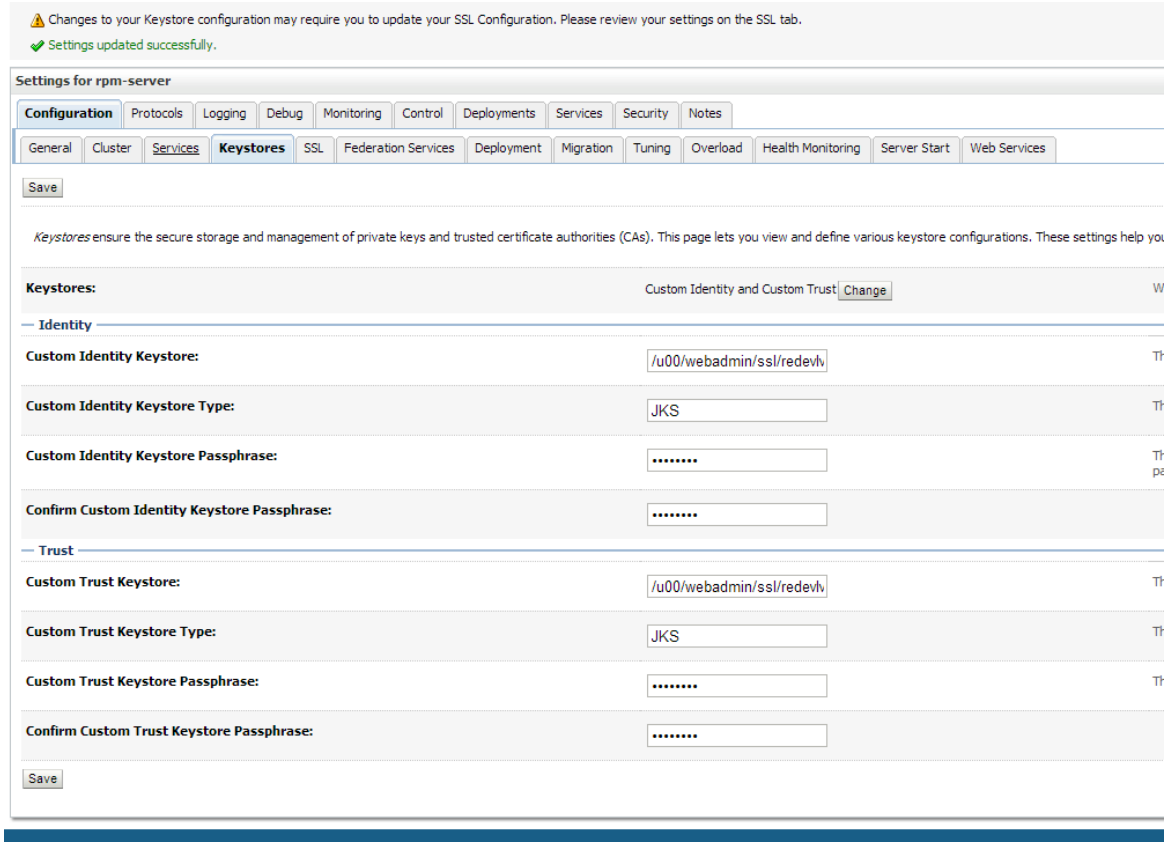
      If you chose **Java Standard Trust** as your keystore, specify the password defined when creating the keystore. Confirm the password.

      If you chose **Custom Trust [Recommended]**, define the following attributes:
      – **Custom Trust Keystore**: The fully qualified path to the trust keystore.

- **Custom Trust Keystore Type**: The type of the keystore. Generally, this attribute is JKS; if left blank, it defaults to JKS.
- **Custom Trust Keystore Passphrase**: The password you will enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore so whether or not you define this property depends on the requirements of the keystore.

h. Click Save.

i. To activate these changes, in the Change Center of the Administration Console, click Activate Changes.
Not all changes take effect immediately—some require a restart.



For more details See "Configure Keystores" in the *Administration Console Online Help*.

2. Set SSL configuration options for the private key alias and password in the WebLogic Server Administration Console.

a. In the Change Center of the Administration Console, click Lock & Edit.

b. In the left pane of the Console, expand Environment and select Servers.

c. Click the name of the server for which you want to configure the identity and trust keystores.

d. Select Configuration > SSL.

e. In the Identity and Trust Locations, defaults to Keystores.

f. In the Private Key Alias, type the string alias used to store and retrieve the server's private key.

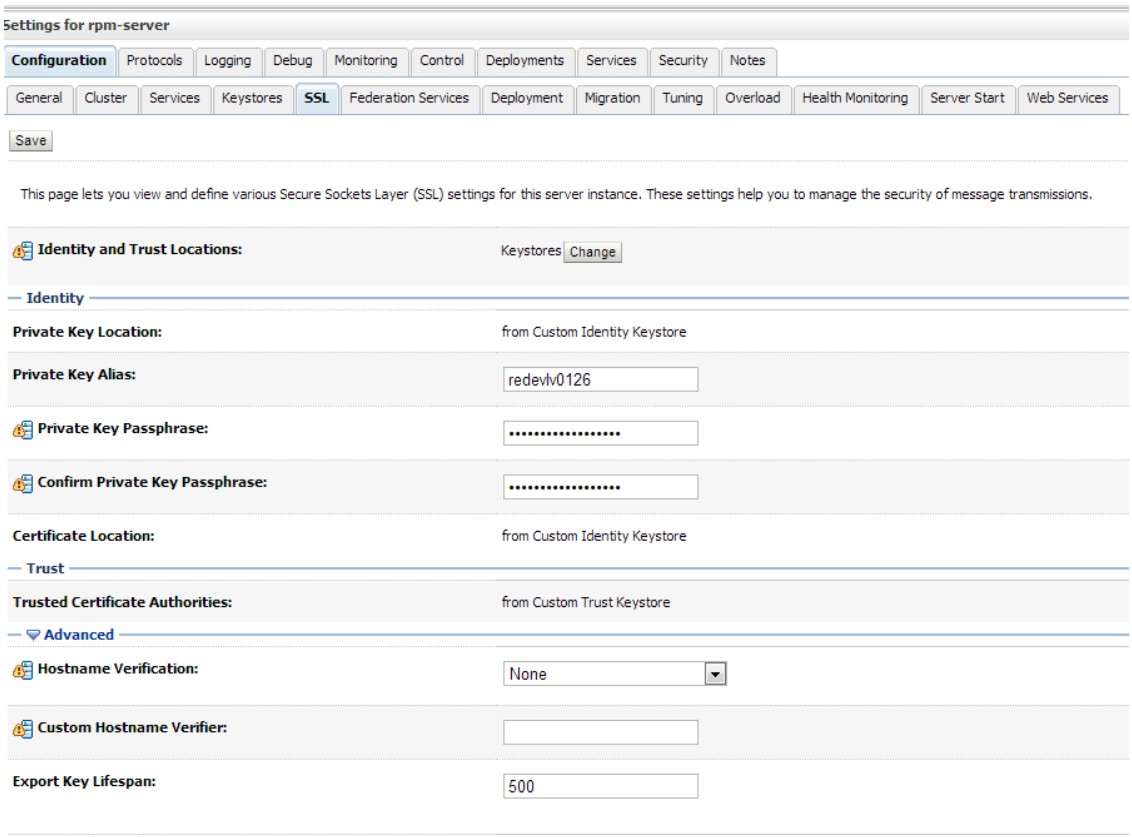
g. In the Private Key Passphrase, provide the keystore attribute that defines the passphrase used to retrieve the server's private key.

h. Save the changes.

i. Click on Advanced Section of SSL tab.

j. In the Hostname Verification, select as None. This specifies to ignore the installed implementation of the weblogic.security.SSL.HostnameVerifier interface (this interface is generally used when this server is acting as a client to another application server).

k. Save the changes



For more details see "Configure SSL" in the *Administration Console Online Help*.

Verify SSL Connections

All the server SSL attributes are dynamic; when modified via the Console, they cause the corresponding SSL server or channel SSL server to restart and use the new settings for new connections. Old connections will continue to run with the old configuration. To ensure that all the SSL connections exist according to the specified configuration, you must reboot WebLogic Server.

Use the **Restart SSL** button on the Control: Start/Stop page to restart the SSL server when changes are made to the keystore files and need to be applied for subsequent connections without rebooting WebLogic Server.

Upon restart you can see similar entries in the log.

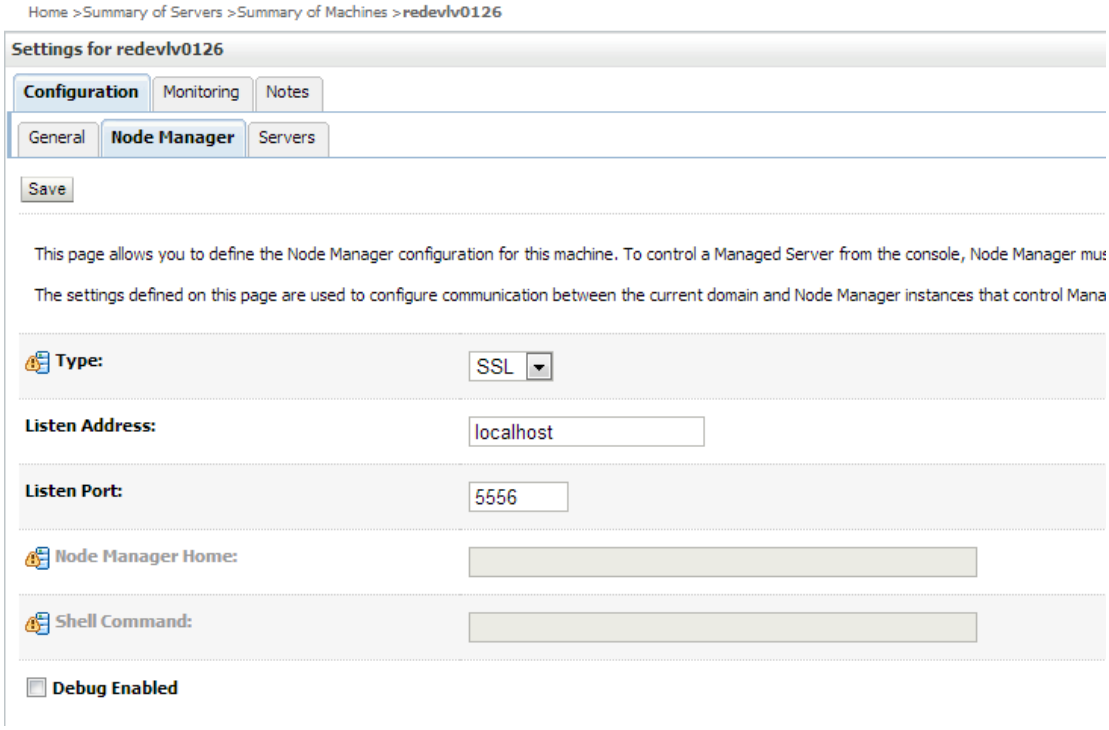
```
<Mar 11, 2013 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000365> <Server state
changed to RESUMING>
<Mar 11, 2013 5:18:27 AM CDT> <Notice> <Server> <BEA-002613> <Channel
"DefaultSecure" is now listening on 10.141.15.214:57002 for protocols iiops, t3s,
ldaps, https.>
<Mar 11, 2013 5:18:27 AM CDT> <Notice> <Server> <BEA-002613> <Channel
"DefaultSecure[1]" is now listening on 127.0.0.1:57002 for protocols iiops, t3s,
ldaps, https.>
<Mar 11, 2013 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000329> <Started
WebLogic Admin Server "AdminServer" for domain "APPDomain" running in Production
Mode>
<Mar 11, 2013 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000365> <Server state
changed to RUNNING>
<Mar 11, 2013 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000360> <Server
started in RUNNING mode>
```

> **Note:** For complete security of the WebLogic Server, it is recommended to secure both **Administration** as well the **Managed server** where application is being deployed. You can choose to disable the non-SSL ports (HTTP). It is highly recommended to secure the Node Manager. The steps to secure Node Manager as provided in the following section.

Securing Nodemanager with SSL Certificates

1. Navigate to **<BEA_HOME>/wlserver_10.3/common/nodemanager** and take a backup of nodemanager.properties
2. Add similar entry to **nodemanager.properties**.
   - **KeyStores**=CustomIdentityAndCustomTrust
   - **CustomIdentityKeyStoreFileName**=/u00/webadmin/ssl/redevlv0126.keystore
   - **CustomIdentityKeyStorePassPhrase**=[password to keystore, this will get encrypted]
   - **CustomIdentityAlias**=redevlv0126
   - **CustomIdentityPrivateKeyPassPhrase**=[password to keystore, this will get encrypted]
   - **CustomTrustKeyStoreFileName**=/u00/webadmin/ssl/redevlv0126.keystore
   - **SecureListener**=true
3. Login to WebLogic console, navigate to **Environment** > **Machines**. Select the nodemanager created already and navigate to **Node Manager** tab. In the Change Center, click **Lock and Edit**.

For **Type**, select SSL and save and activate.



4. After activating the changes, bounce the entire WebLogic Domain for changes to take effect. Verify that the nodemanager is reachable in the **Monitoring** tab after the restart.

Using Secured LDAP

The application can communicate with the LDAP server on a secured port. It is recommended that you use secured an LDAP server for security.

Refer to Configuring Secure Sockets Layer (SSL) in the *Oracle Fusion Middleware Administration Guide* for more details.

In case secure LDAP is used for authentication, it is important to import the certificates used in LDAP server into the JRE of the WebLogic server for SSL handshake.

Example:

```
Set JAVA_HOME and PATH to the JDK being used by WebLogic Domain.
Backup the JAVA_HOME/jre/lib/security/cacerts

/u00/webadmin/product/jdk/jre/lib/security> cp -rp cacerts cacerts_ORIG
```

Import the Root and Intermediary (if required) certificates into the java keystore.

```
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
verisignclass3g3ca -file ~/ssl/Primary.pem -keystore cacerts

/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
oracleclass3g3ca -file ~/ssl/Secondary.pem -keystore cacerts
```

Import the User certificate from LDAP server into the java keystore.

```
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
redevlv0126 -file ~/ssl/cert.cer -keystore cacerts
```

Note: The default password for the JDK keystore is changeit.

The deployed application should be able to communicate with LDAP on the SSL port after a successful SSL handshake.

Batch Setup for SSL Communication

Batch programs communicate with Java applications deployed in WebLogic. The communication needs to have an SSL handshake with the deployed application.

Example:

```
/u00/webadmin/product/jdk/jre/lib/security> cp -rp cacerts cacerts_ORIG

/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
verisignclass3g3ca -file ~/ssl/Primary.pem -keystore cacerts

/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
oracleclass3g3ca -file ~/ssl/Secondary.pem -keystore cacerts

/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
redevlv0126 -file ~/ssl/cert.cer -keystore cacerts
```

Note: The default password for the JDK keystore is changeit.

H

Appendix: Certificate Import Topology

Implementation of SSL into the Retail deployment is driven by mapping the SSL certificates and wallets to various participating components in the topology. The table below describes the trust stores to be updated while confirming the certificates imported into middleware and repository of Retail applications. Please ensure you have updated the given trust stores with the signed (either self signed or issued by certifying authority) certificates.

	Java app-host		Forms app-host		RIB app-host		BIPublisher-host		OID-host	Client-host	
Certificates	**Java app - Managed server**	**Java app-JAVA cacerts**	**Forms app - Managed server**	**Forms app-JAVA cacerts**	**RIB app-Managed server**	**RIB app-JAVA cacerts**	**BIPublisher-Managed server**	**BIPublisher-JAVA cacerts**	**Wallet**	**Browser**	**Client-JAVA cacerts**
appserver.cer	Yes	No	No	No	No	No	No	No	No	No	No
approot.cer	Yes	Yes	No	No	No	Yes	No	Yes	Yes	Yes	Yes
frmserver.cer	No	No	Yes	No	No	No	No	No	No	No	No
frmroot.cer	No	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes
ribserver.cer	No	No	No	No	Yes	No	No	No	No	No	No
ribroot.cer	No	Yes	No	No	Yes	Yes	No	No	No	Yes	Yes
biserver.cer	No	No	No	No	No	No	Yes	No	No	No	No
biroot.cer	No	Yes	No	Yes	No	No	Yes	Yes	No	Yes	Yes
oidcer.cer	No	No	No	No	No	No	No	No	Yes	No	No
oidroot.cer	No	Yes	No	Yes	No	No	No	Yes	Yes	Yes	Yes

I

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), Oracle Retail Sales Audit (ReSA). Optional: Oracle Retail Fiscal Management (ORFM)

   > **Note:** ORFM is an optional application for RMS if you are implementing Brazil localization.

2. Oracle Retail Service Layer (RSL)
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 Central Office (ORCO)
10. Oracle Retail Returns Management (ORRM)
11. Oracle Retail Back Office (ORBO) or Back Office with Labels and Tags (ORLAT)
12. 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.

13. Oracle Retail Predictive Application Server (RPAS)
14. Oracle Retail Demand Forecasting (RDF)
15. Oracle Retail Category Management (CM)
16. Oracle Retail Replenishment Optimization (RO)
17. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)

18. Oracle Retail Regular Price Optimization (RPO)
19. Oracle Retail Merchandise Financial Planning (MFP)
20. Oracle Retail Size Profile Optimization (SPO)
21. Oracle Retail Assortment Planning (AP)
22. Oracle Retail Item Planning (IP)
23. Oracle Retail Item Planning Configured for COE (IP COE)
24. Oracle Retail Advanced Inventory Planning (AIP)
25. Oracle Retail Integration Bus (RIB)
26. Oracle Retail Point-of-Service (ORPOS)
27. Oracle Retail Markdown Optimization (MDO)
28. Oracle Retail Clearance Optimization Engine (COE)
29. Oracle Retail Analytic Parameter Calculator for Markdown Optimization (APC-MDO)
30. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
31. Oracle Retail Promotion Intelligence and Promotion Planning and Optimization (PI-PPO)
32. Oracle Retail Analytics
33. Oracle Retail Workspace (ORW)