Oracle® Retail Price Management

Installation Guide
Release 14.0.2
E60689-01

January 2015

ORACLE

Oracle® Retail Price Management Installation Guide, Release 14.0.2

Primary Author: Mourya Pantham

Contributors: Nathan Young

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 14.0.2

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

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

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

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

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

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

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

Preface

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

Audience

This Installation Guide is written for the following audiences:

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

Related Documents

For more information, see the following documents in the Oracle Retail Price Management Release 14.0.2 documentation set:

- *Oracle Retail Price Management Release Notes*
- *Oracle Retail Price Management Operations Guide*
- *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, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 14.0) or a later patch release (for example, 14.0.2). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

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

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

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

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-**02** is an updated version of a document with part number E123456-**01**.

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

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

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.

Implementation Capacity Planning

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

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

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

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

Requesting Infrastructure Software

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

Check Supported Database Server Requirements

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

Supported on:	Versions Supported:
Database Server OS	OS certified with Oracle Database 11gR2 (11.2.0.4) Enterprise Edition. Options are: ▪ Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.4) with the following specifications: **Components:** ▪ Oracle Partitioning ▪ Examples CD (Formerly the companion CD) **Oneoff Patches:** ▪ 18465025: MERGE REQUEST ON TOP OF 11.2.0.4.0 FOR BUGS 18016963 18302329. **Other Components:** ▪ Perl interpreter 5.0 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.

Supported on:	Versions Supported:
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.7) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.6) ▪ Oracle Identity Management 11g Release 1 (11.1.1.7) ▪ ADF 11.1.1.7 ▪ Oracle Enterprise Manager ▪ **Note:** Oracle Internet Directory (OID) is the supported LDAP directory for Oracle Retail products. For alternate LDAP directories, refer to Oracle WebLogic documentation set. **Java:** ▪ JDK 1.7+ 64 bit **IMPORTANT**: If there is an existing WebLogic installation on the server, you must upgrade it to WebLogic 10.3.6. All middleware components associated with WebLogic server should be upgraded to 11.1.1.7. **Optional (required for SSO**) ▪ Oracle WebTier 11g (11.1.1.7) Oracle Access Manager 11g Release 1 (11.1.1.7) **Note:** A separate WebLogic 10.3.5 installation is required for Oracle Access Manager 11g. ▪ Oracle Access Manager Agent (WebGate) 11g Release 1 (11.1.1.7)

Check Single Sign-On Requirements

If RPM will not be deployed in a Single Sign-On environment, skip this section.

If Single Sign-On is to be used, verify the Oracle Identity Management 11gR1 version 11.1.1.7 has been installed along with the components listed in the above Application Server requirements section. Verify the Oracle WebTier with Webgate Server is registered with the Oracle Access Manager 11gR1.

Check Supported Client PC and Web Browser Requirements

Requirement	Version
Operating system	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	1.7+
Browser	Microsoft Internet Explorer 9 or 11 Mozilla Firefox 24+ **Note:** Other Oracle Merchandising applications may not have the same levels of browser certification.

Check Oracle Retail Software Dependencies

The database portion of the RMS 14.0.2 application must be installed prior to installing RPM.

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)/Oracle Retail Trade Management (RTM)/Oracle Retail Sales Audit (ReSA)	14.0.2
Oracle Retail Allocation	14.0.2
Oracle Retail Store Inventory Management (SIM)	14.0.2
Oracle Retail POS Suite	14.0.2

Supported Oracle Retail Integration Technologies

Requirement	Version
Oracle Retail Integration Bus (RIB)	14.0.2
Oracle Retail Service Backbone (RSB)	14.0.2
Oracle Retail Service Layer (RSL)	14.0.2

Check Third-Party Software Dependencies

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

UNIX User Account Privileges to Install the Software

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

For example, oretail.

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

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 11.2.0.4 RAC database. When using a RAC database, all JDBC connections should be configured to use THIN connections rather than OCI connections.

Clustering for WebLogic Server 10.3.6 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 11.2.0.4 Oracle Internet Directory database with the WebLogic 10.3.6 cluster. It is suggested that a Web Tier 11.1.1.7 installation be configured to reflect all application server installations if SSO will be utilized.

References for Configuration

- Oracle Fusion Middleware High Availability Guide 11g Release 1 (11.1.1) Part Number E10106-09
- Oracle Real Application Clusters Administration and Deployment Guide 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 14.0.2 is a prerequisite of the RPM 14.0.2 installation.

4

Application Installation Tasks

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

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

Steps to Create the Domain with ADF Libraries and Enterprise Manager

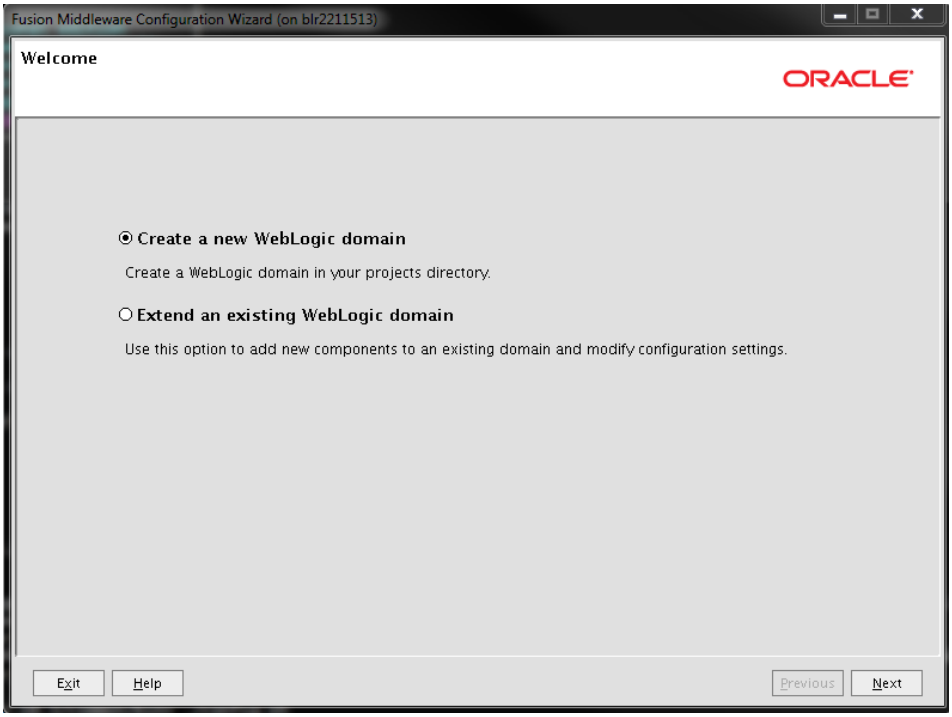
1. Set the required environment variables

   ```
   export JAVA_HOME=<JDK_HOME>
   export ORACLE_HOME=<WLS_HOME>
   ```

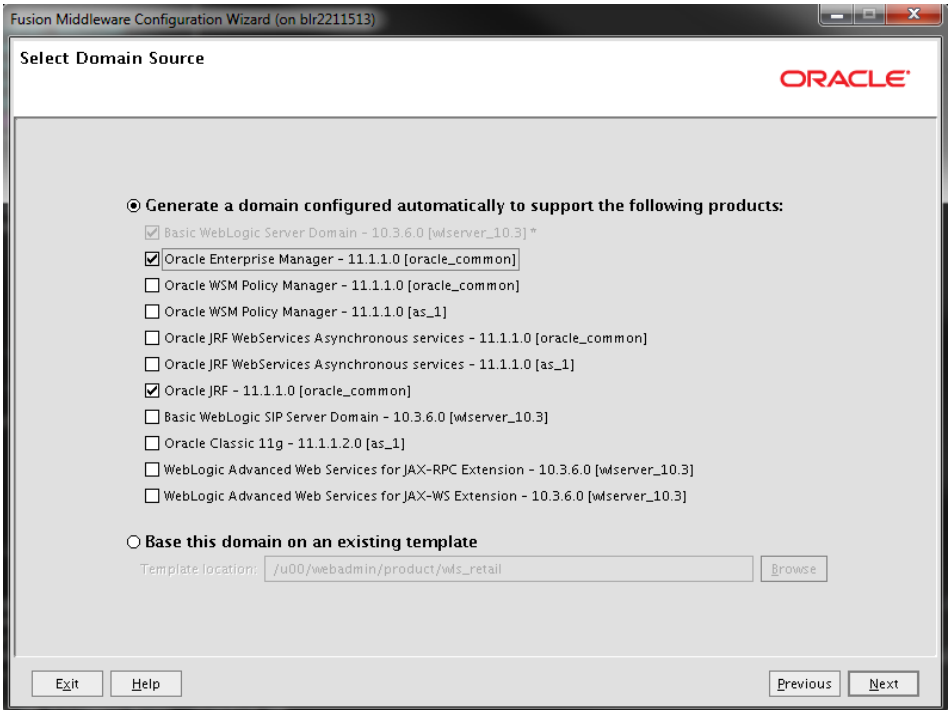
   (Example:/u00/webadmin/product/wls_retail)

   ```
   export PATH=$JAVA_HOME/bin:$PATH
   export
   ORACLE_HOME=<WLS_HOME>/oracle_common(Example:/u00/webadmin/product/wls_retail/
   oracle_common)
   ```

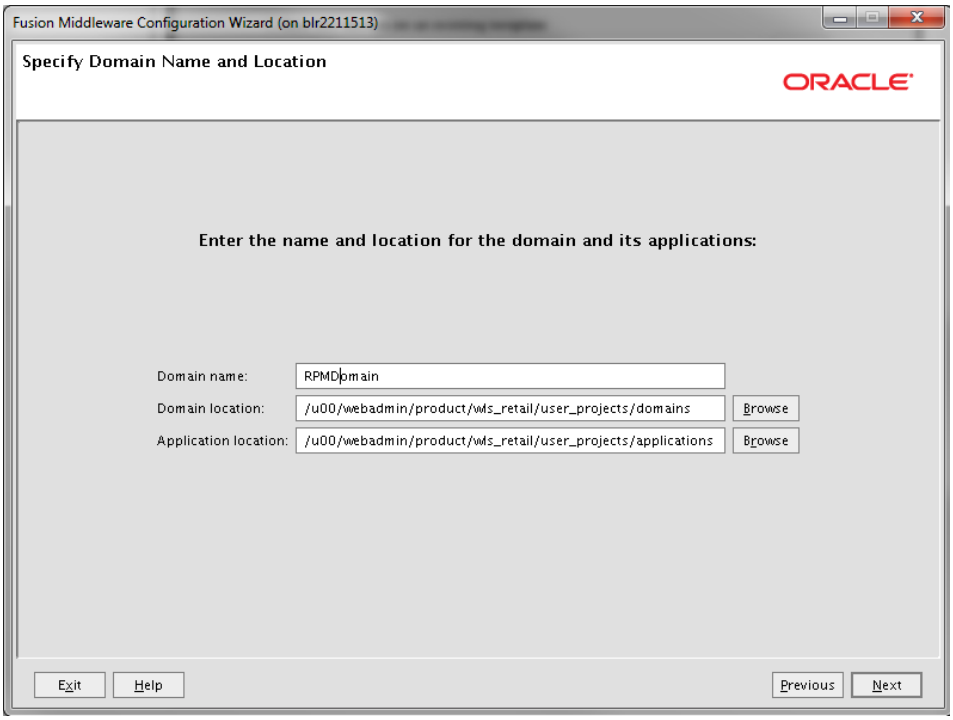
2. Change directories to $ORACLE_HOME/wlserver_10.3/common/bin and run the config.sh scripts to create the new RPMdomain with Enterprise Manager. The following screen is displayed. Select the default and Click on Next



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



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

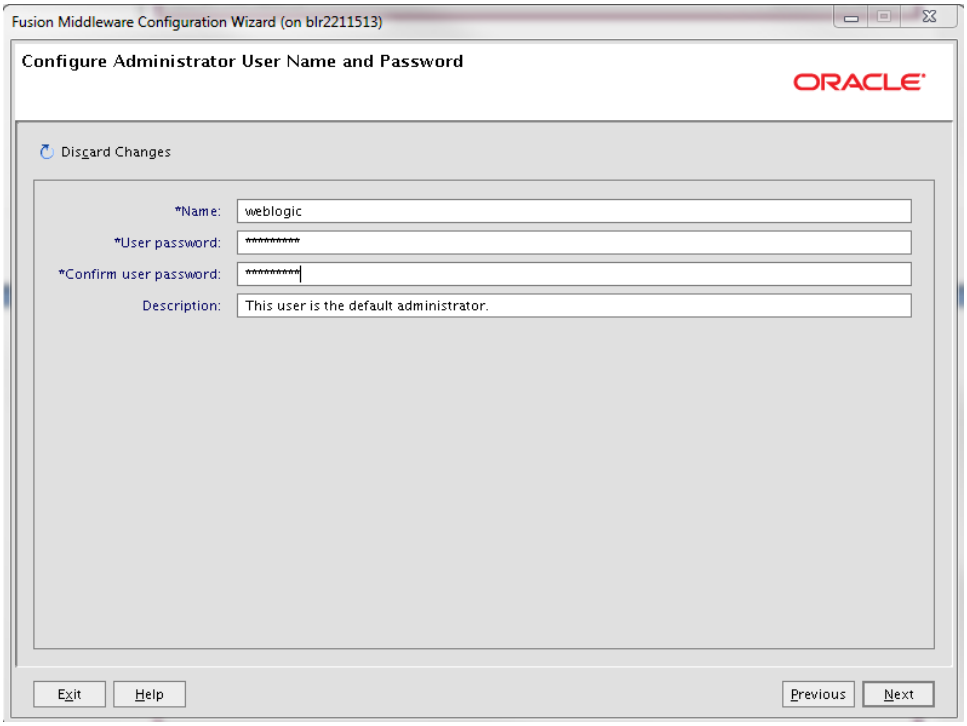


5. Enter 'User password' value and 'Confirm user password' value (same as user password).

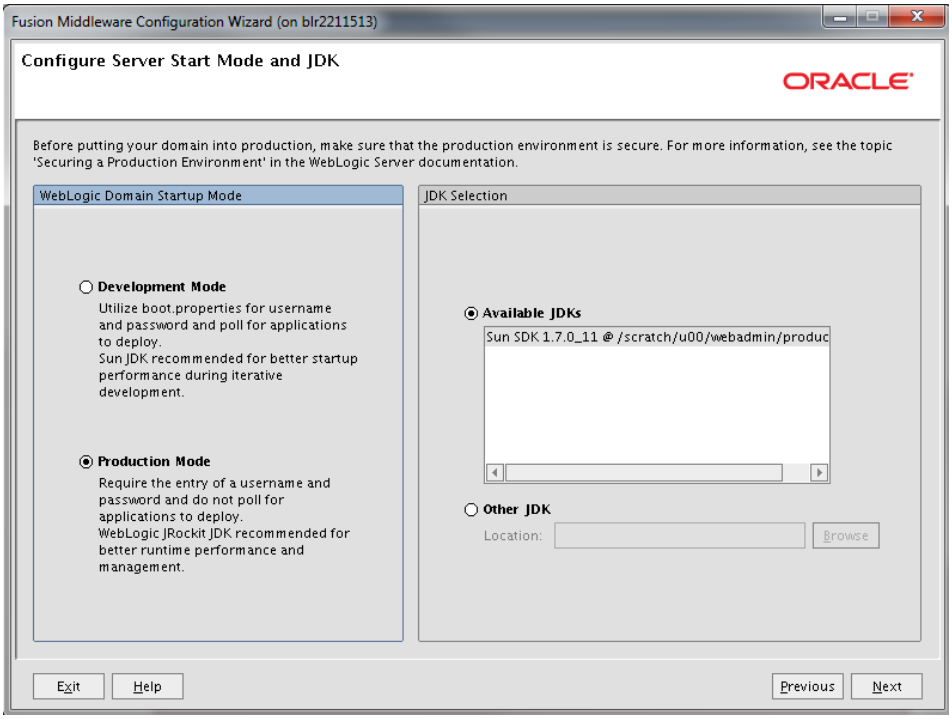
   User password=<password>

   Confirm user password=<password>

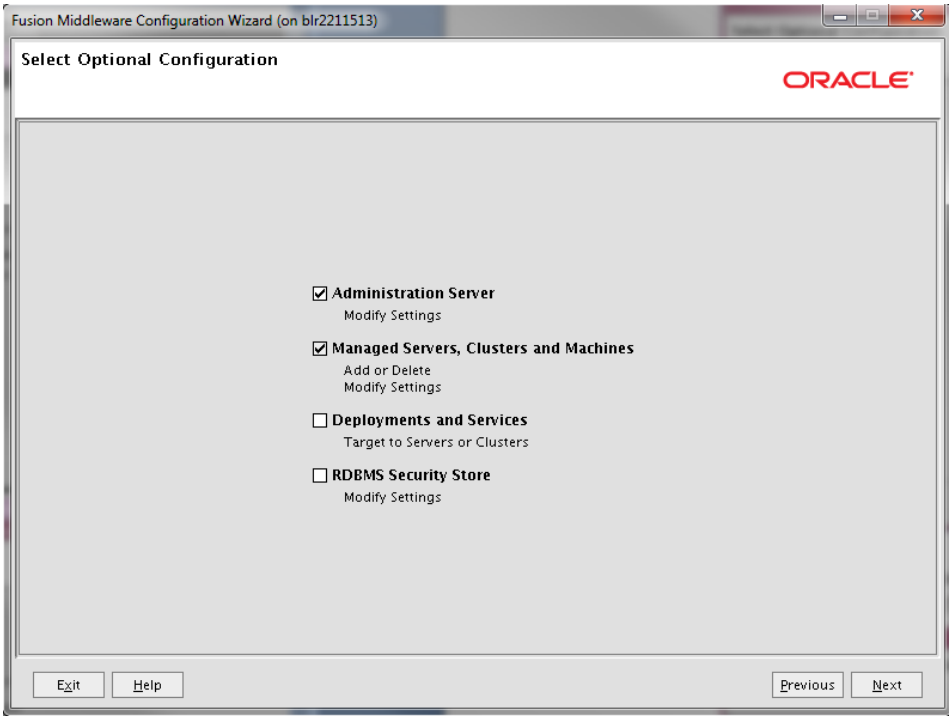
   Click **Next**.



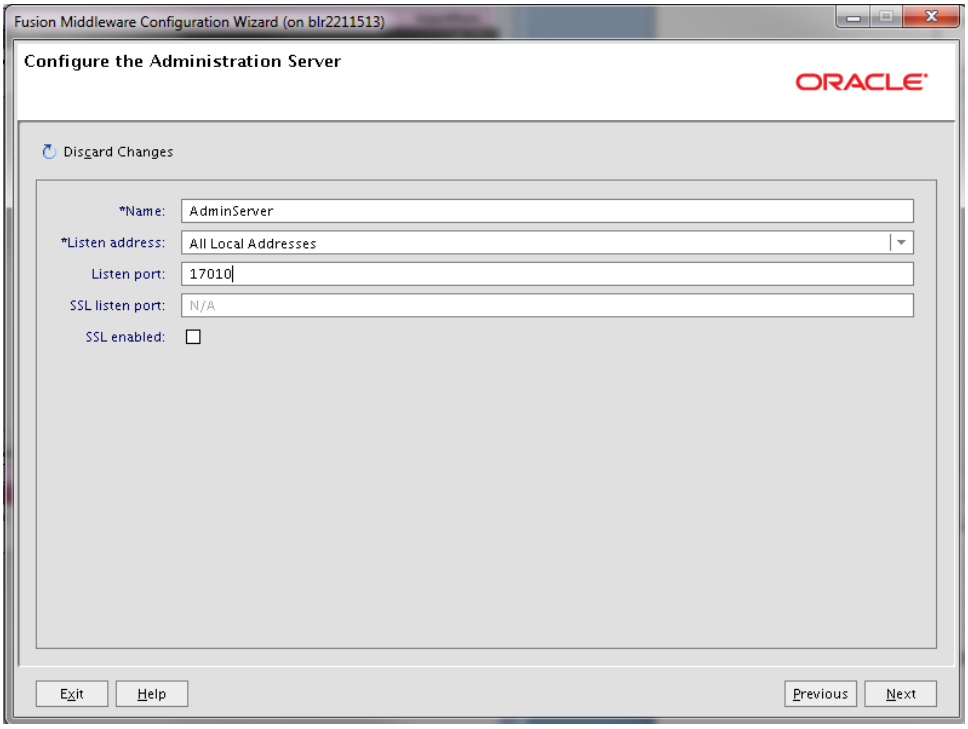
6. Select Production Mode for WebLogic domain Startup Mode. Click **Next.**



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

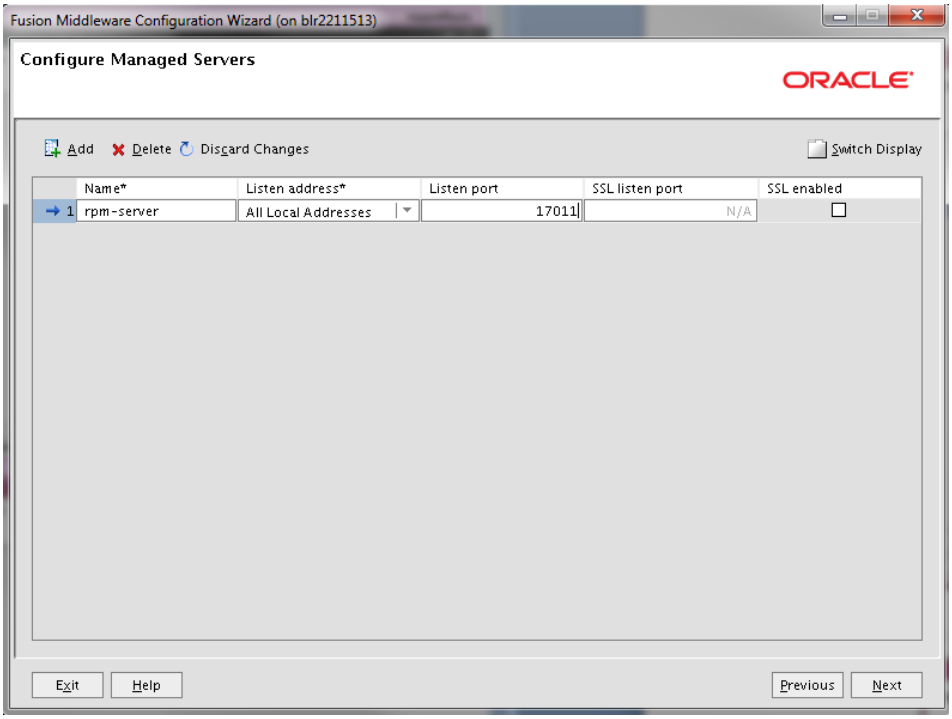


8. Enter the Listen port and click **Next**.
   - Listen port: 17010 (This port must be an open port on the server)

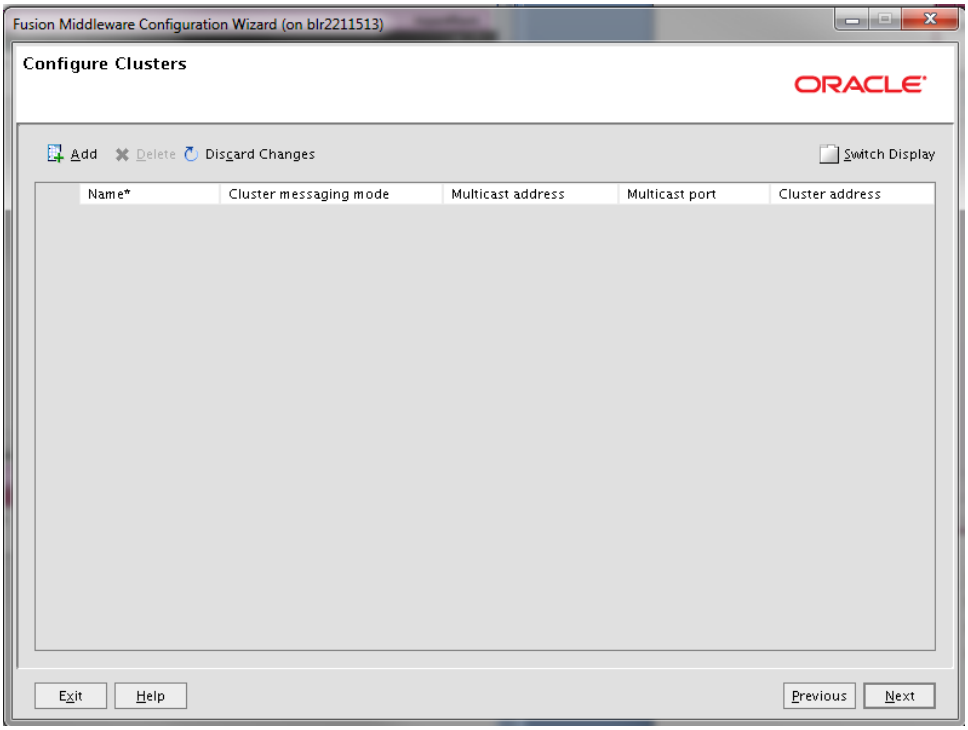


9. Click **Add** and provide Name and Listen Port for the managed server.
   - Name: rpm-server (This is your managed server name)
   - Listen port: 17011 (This port must be an open port on the server)

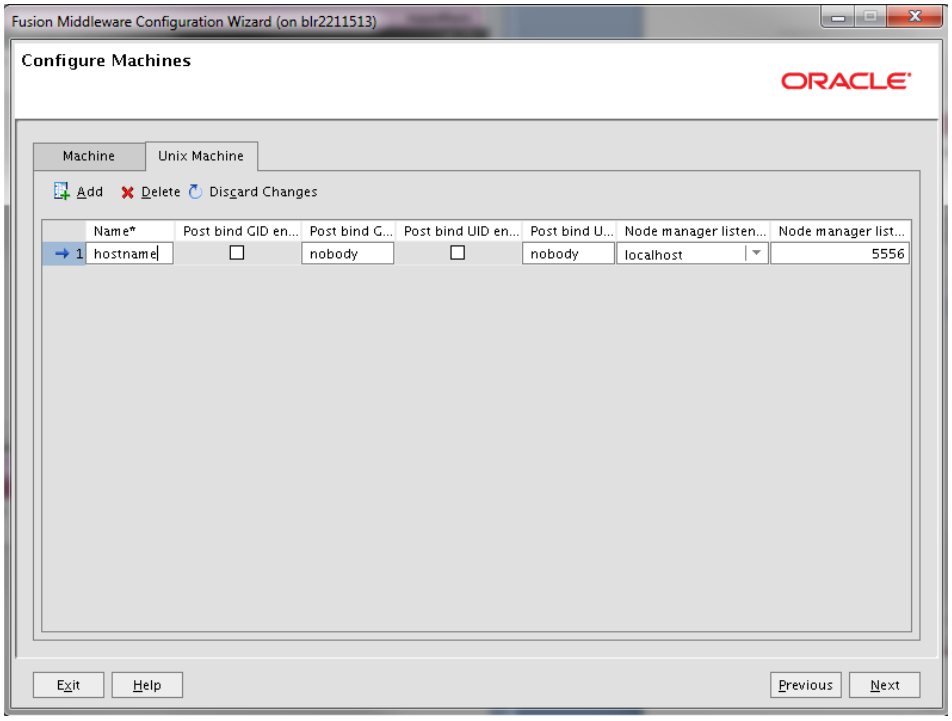
10. Click **Next**.



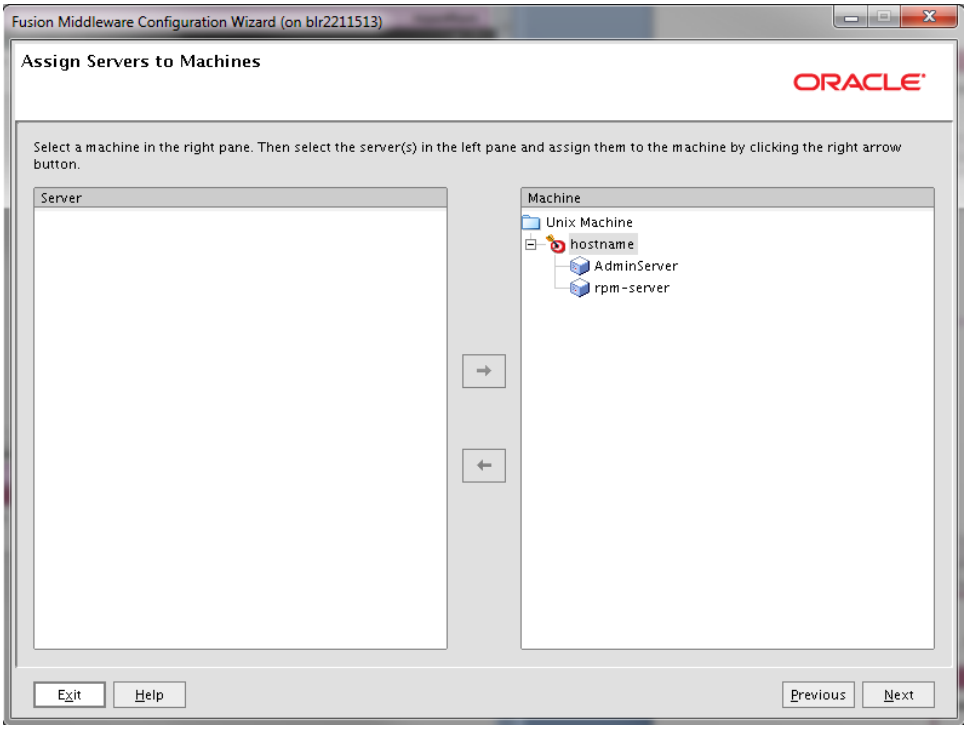
11. Click **Next**.



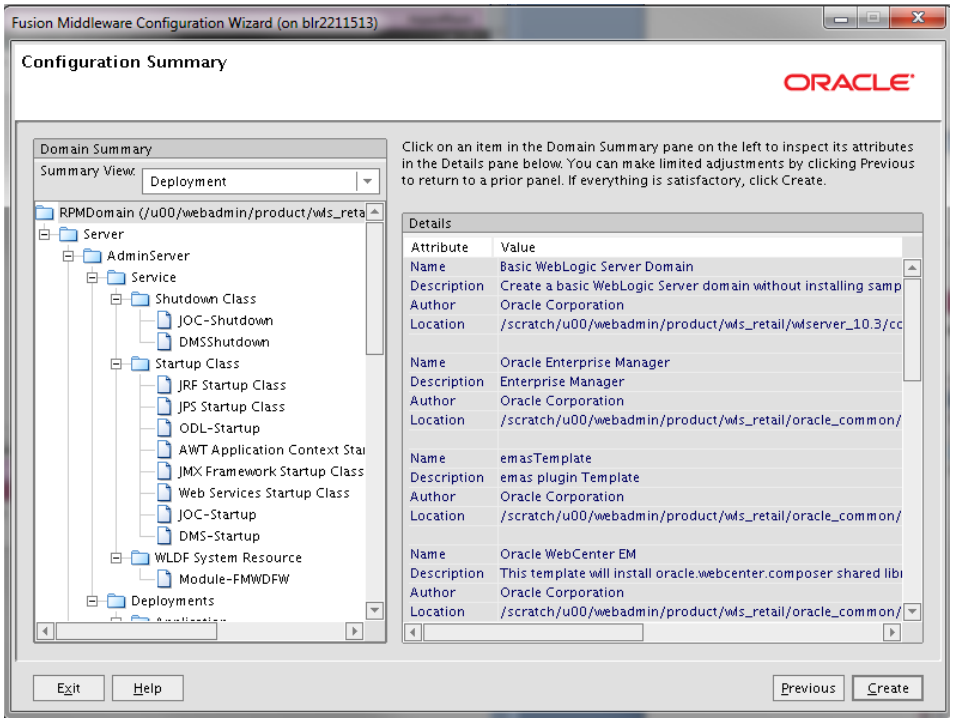
12. In 'Unix Machine' tab, Click Add and provide Name and an open port for Node manager listen port as shown below.
    - Name: <hostname> (This can be any name or usually your hostname)
    - Listen port: 5556 (This port must be an open port on the server)



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



14. Click **Create**.



Update the WebLogic.policy

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

> **Note:** If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semi colon. Also, the AdminServer must be restarted for these changes to take effect.
>
> **Note:** <WEBLOGIC_DOMAIN_HOME> in the example below is the full path of the WebLogic domain; <managed_server> is the RPM managed server created.

```
grant codeBase
"file:<WEBLOGIC_DOMAIN_HOME>/servers/<managed_server>/tmp/_WL_user/<context_ro
ot>/-" {
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/RPMdomain/servers
/rpm-server/tmp/_WL_user/rpm14/-" {
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/RPMdomain/servers
/rpm-server/cache/EJBCompilerCache/-" {
permission java.security.AllPermission;
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete";
permission oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};
```

Start the Node Manager

1. Start up the nodemanager. Edit the nodemanager.properties file at the following location with the below values:

   $WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties

   - StartScriptEnabled=true
   - StartScriptName=startWebLogic.sh.

2. After making changes to the nodemanager.properties file, NodeManager must be restarted.

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

Start the AdminServer (admin console)

Start up the AdminServer using the RPMdomain/bin/startWebLogic.sh script. With the initial startup you will be asked for the admin user credentials. Once the AdminServer has started up you can create a boot.properties file containing the credentials for the AdminServer to start up without the need to enter the information each time.

An example of the boot.properties would be:

mkdir RPMdomain/servers/AdminServer/security

vi RPMdomain/servers/AdminServer/security/boot.properties

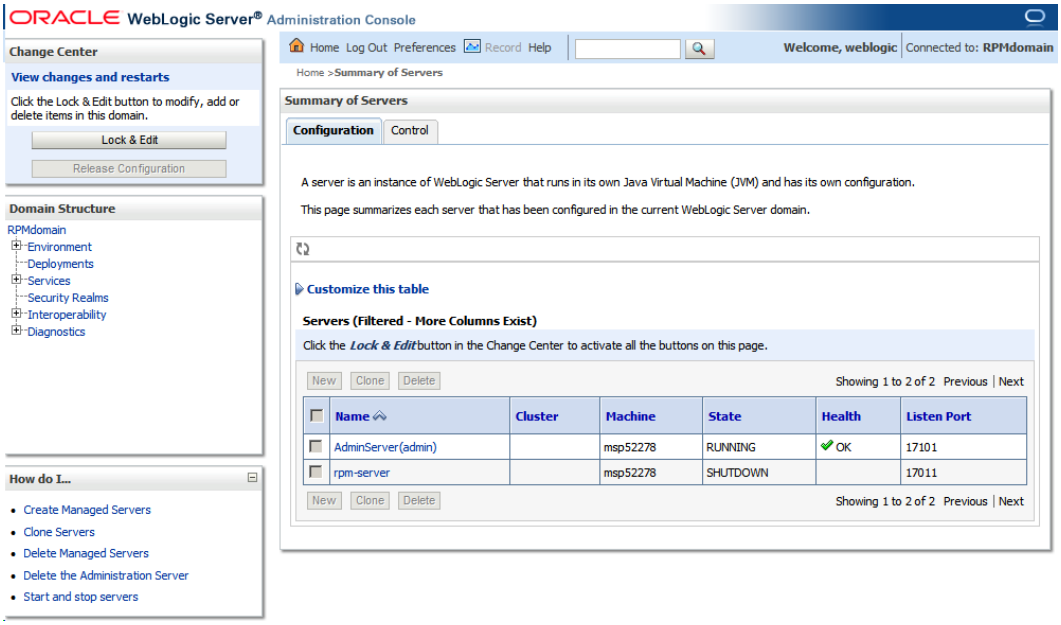
- username=weblogic
- password=<password used at domain creation>

This file will be encrypted after the RPMdomain starts up.

Start the Managed Server

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

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



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

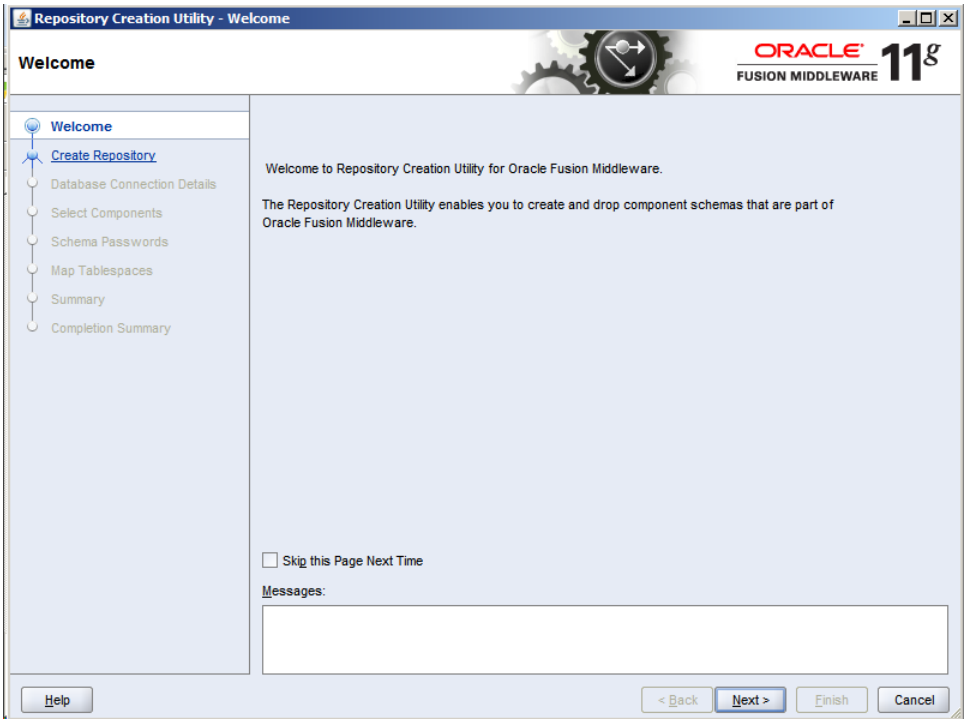
Create Required Schemas with RCU

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

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

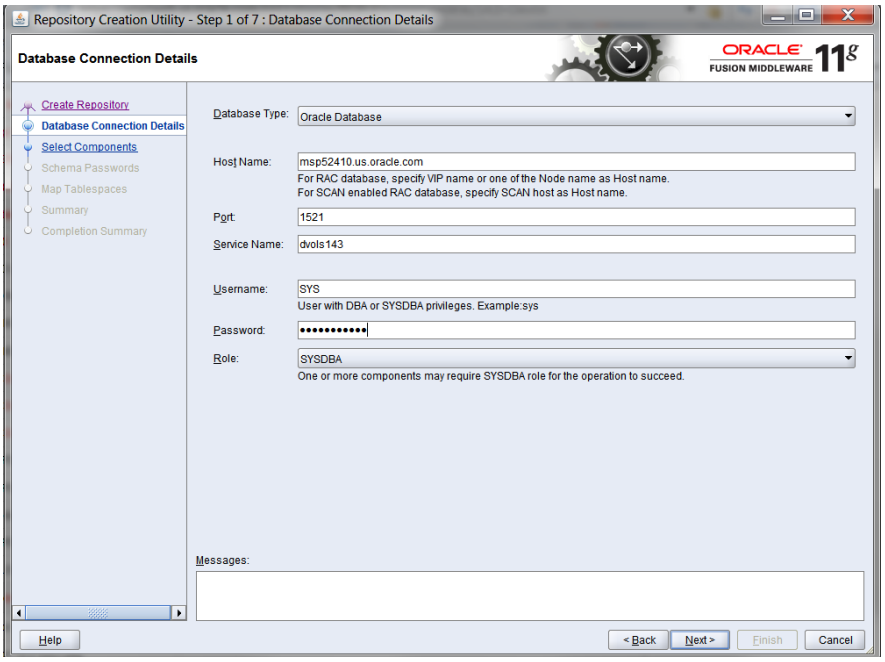
The following steps will show you the creation of the database schemas required:

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

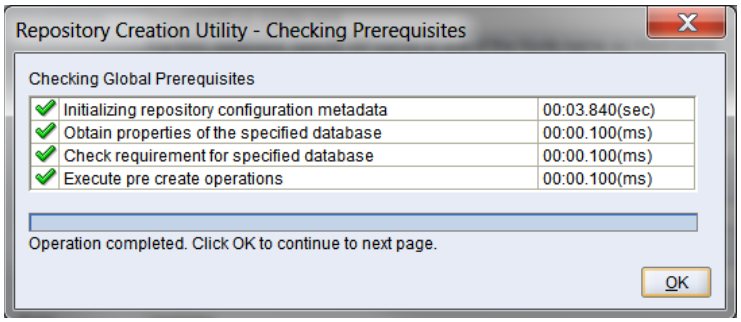


4. Select **Create Repository** and click **Next**.

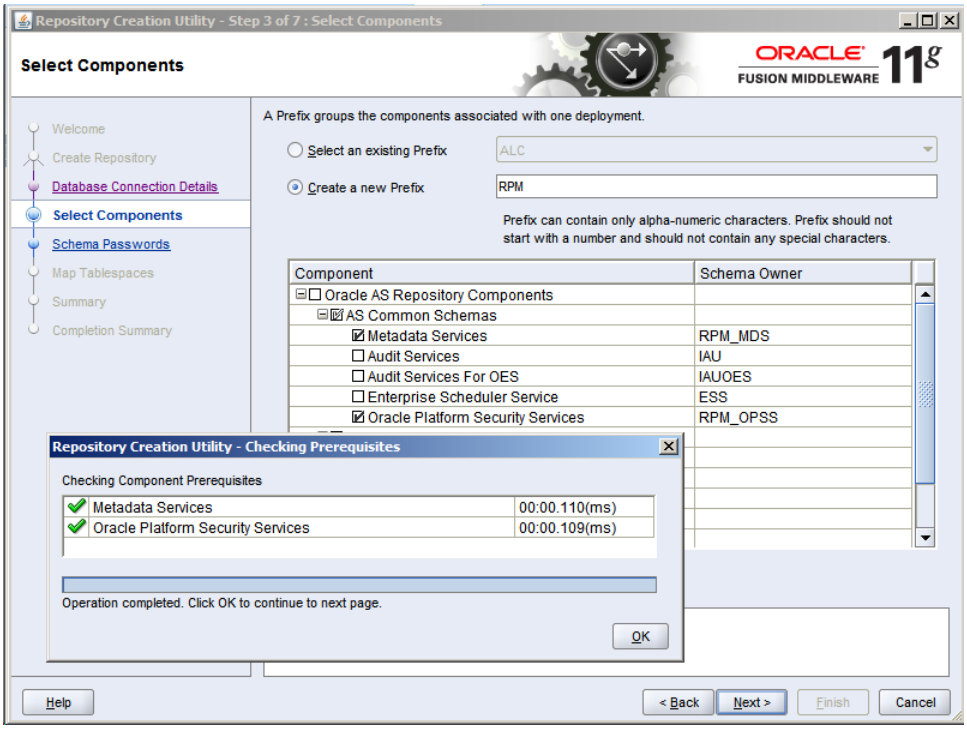
5. Enter all the fields as explained below and click **Next**.
   a. Host Name: Database server host name which Application will use.(example: msp52410.us.oracle.com)
   b. Port: Database port (example: 1521)
   c. Service Name Database name (example: dvols143)
   d. Username: SYS
   e. Password: <SYS password>



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



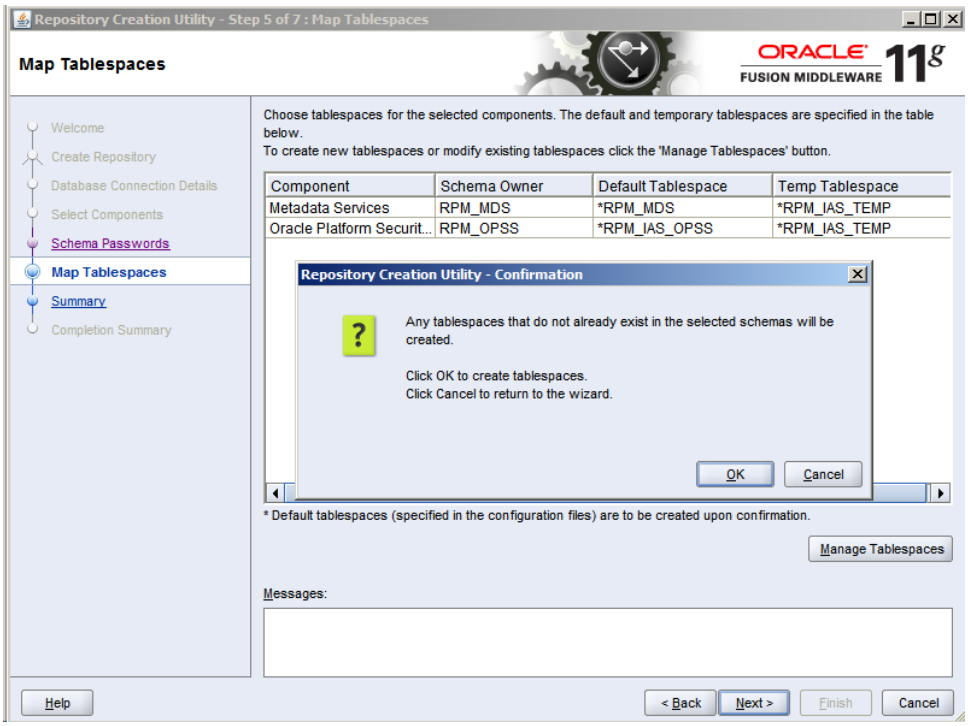
7. Prerequisite requirements are verified and the following screen is displayed. Click **OK**. Expand Oracle AS Common Components and select the Metadata Services and Oracle Platform Security Services checkboxes as shown below. Enter a new prefix if needed (the example uses a prefix of "RPM") and click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



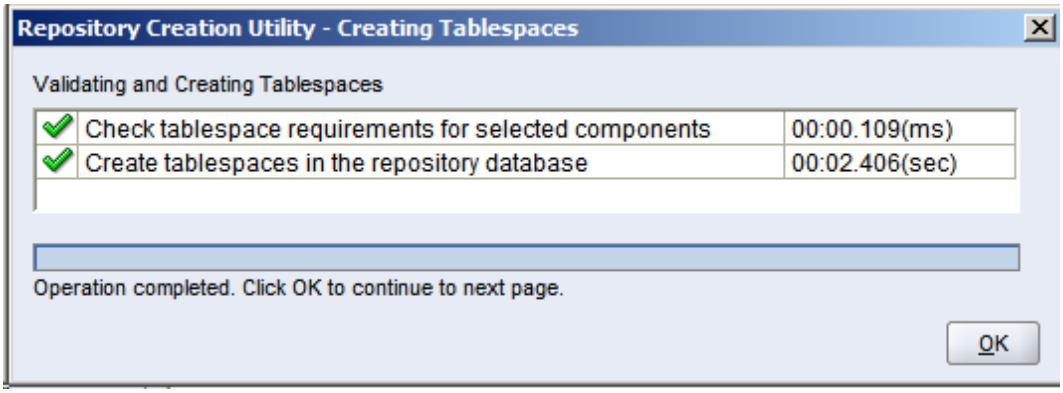
8. Enter and confirm your password and click **Next**.

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

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



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



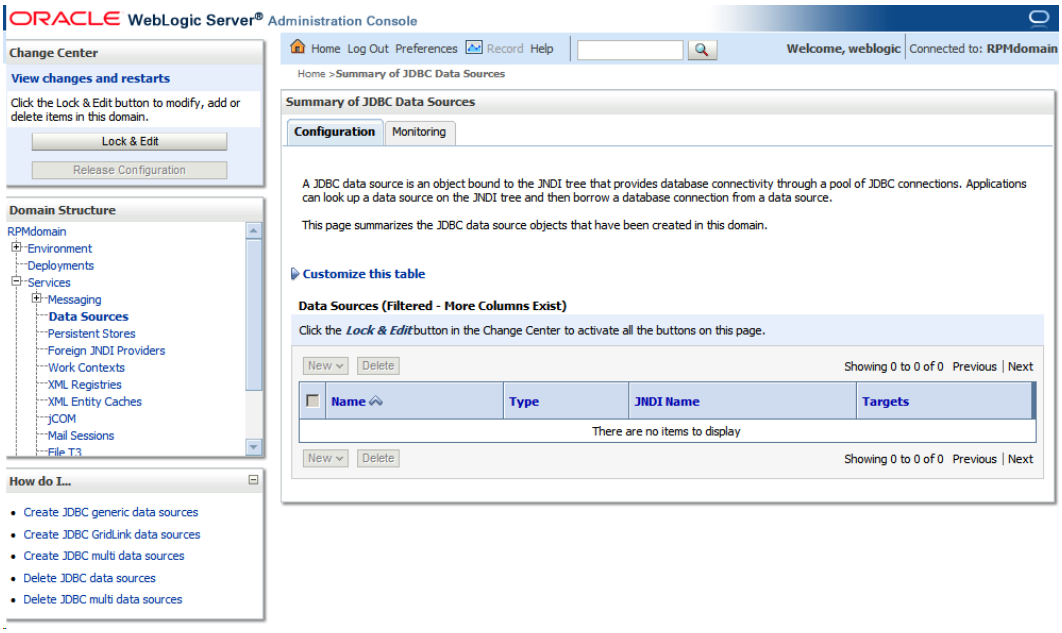
11. Click **Create** to make the schemas.

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

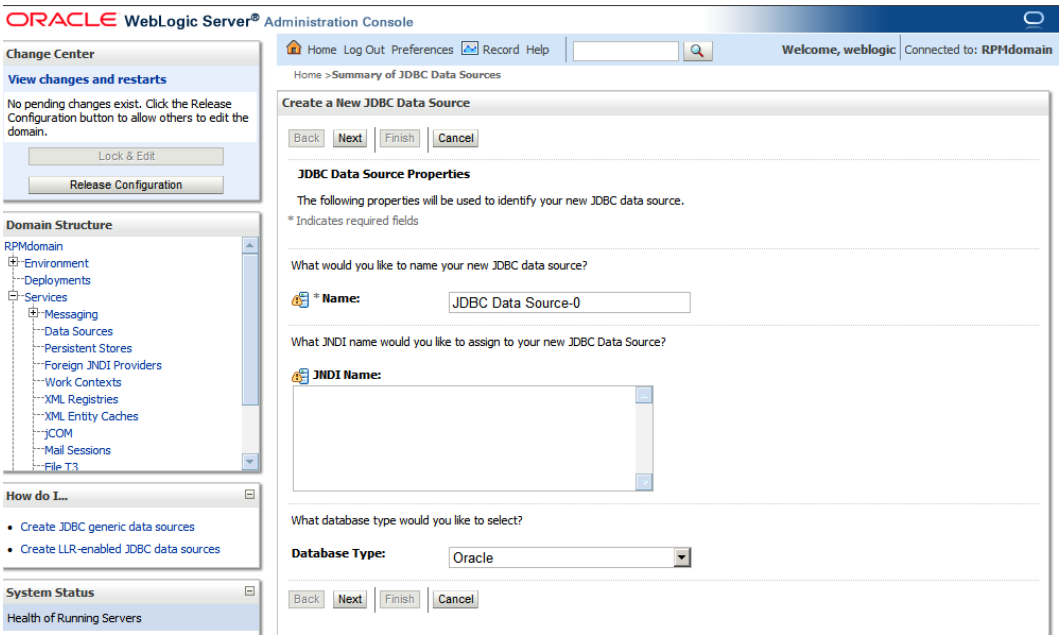
Set up OPSS Schema Data source in WebLogic domain

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

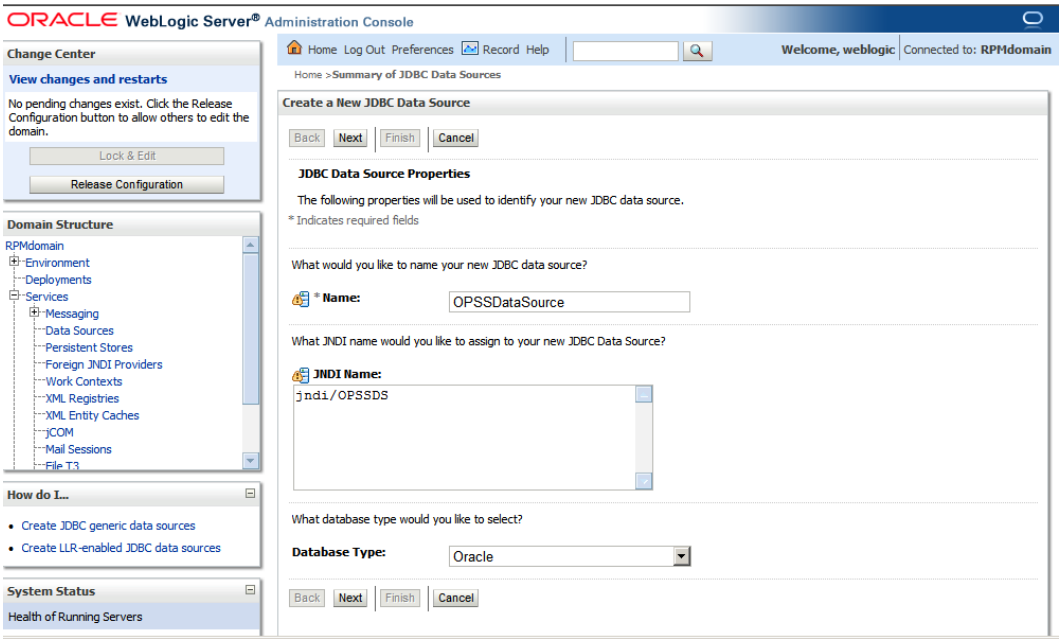
1. Login to the Admin console and go to Services -> Data Sources.



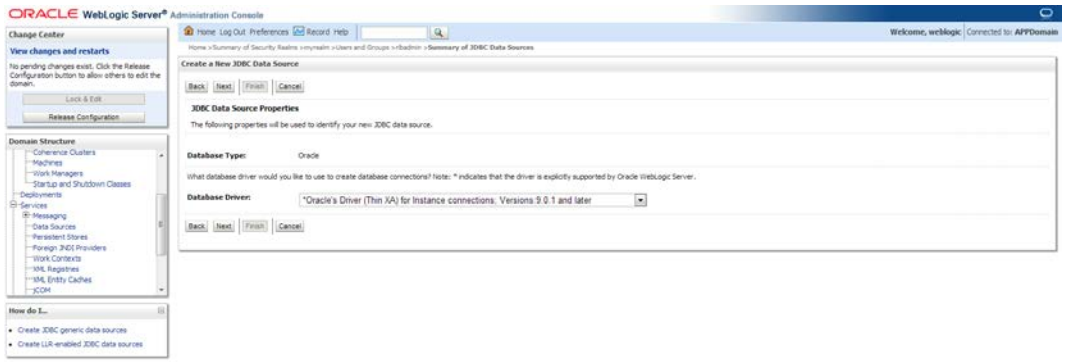
2. Click **Lock & Edit** then click New -> Generic Data Source.



3. Enter the details and click **Next**.
   - Name: OPSSDataSource
   - JNDI Name: jndi/OPSSDS
   - Database Type: Oracle

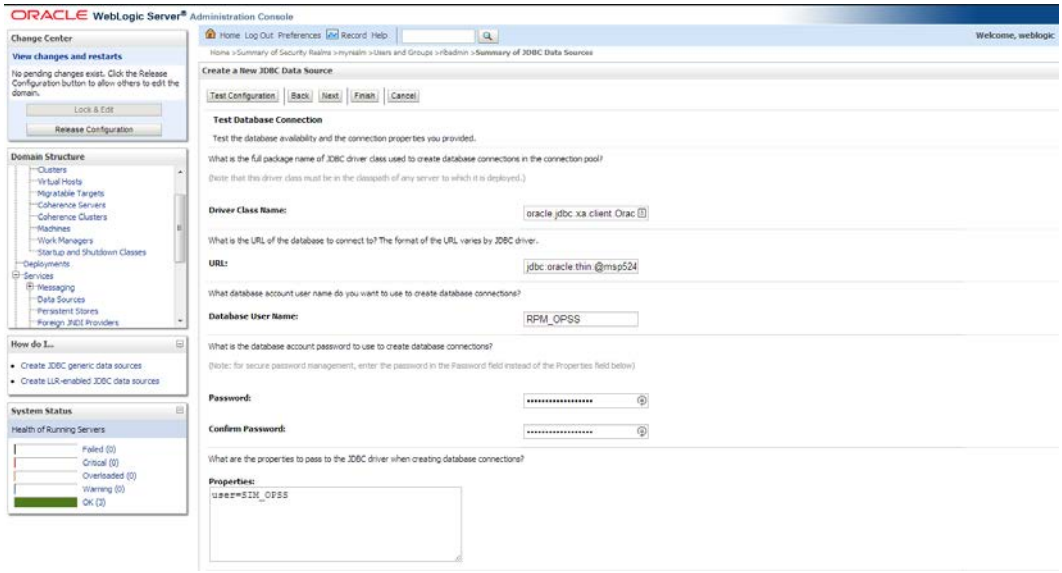


4. Select Oracle's Driver (Thin XA) for Instance connections; Versions: 9.0.1 and later and click **Next**.

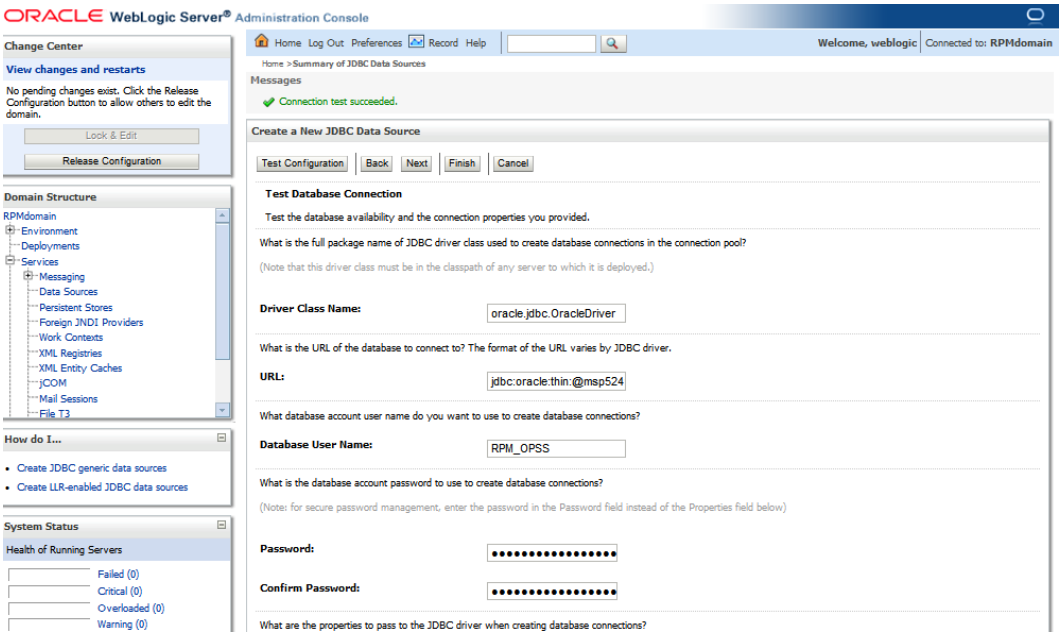


5. Click **Next**.

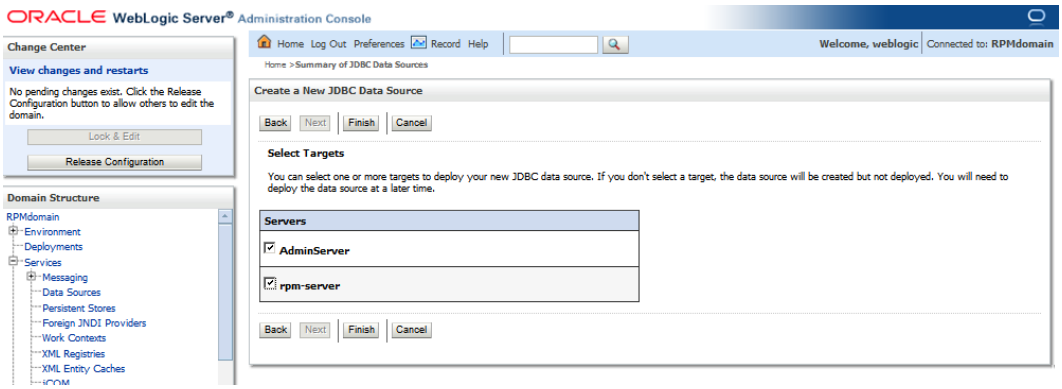
6. Enter the details of the OPSS schema we just created and click **Next**.
   - Database Name: (i.e: dvols143)
   - Host Name: (i.e.: msp52410.us.oracle.com)
   - Port: (i.e.: 1521)
   - Database User Name: RPM_OPSS (This is the OPSS schema which has been created using RCU earlier in this document.)
   - Password: <password> (Password given at the time of OPSS schema creation)



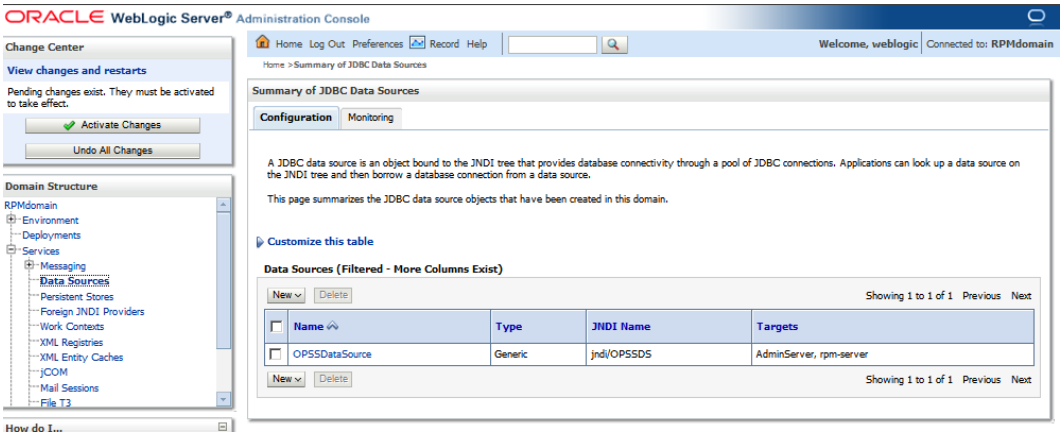
7. Click **Test Configuration** and make sure you can connect to the schema successfully. Click **Next** if you can connect. Click **Back** if it does not connect and check your settings.



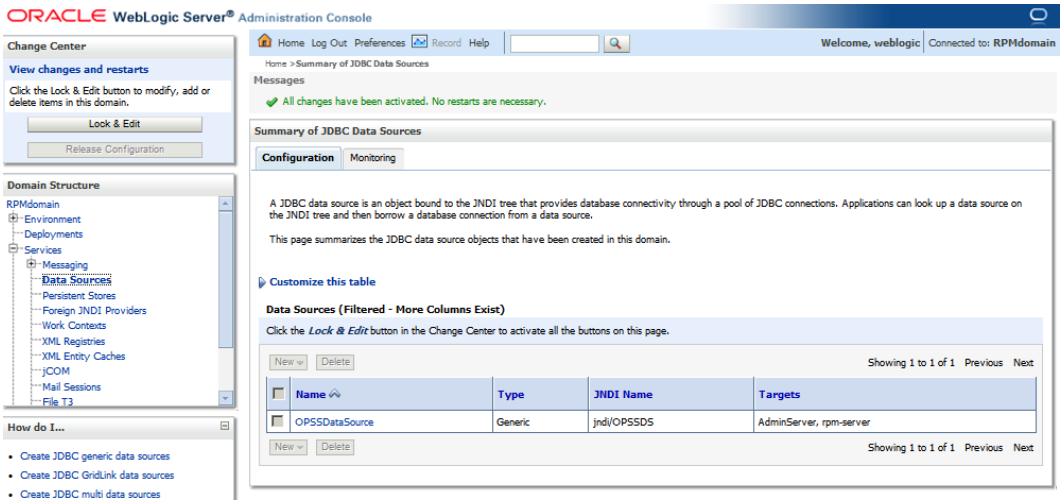
8. Target all the servers (AdminServer & rpm-server) and click **Finish**.



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



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

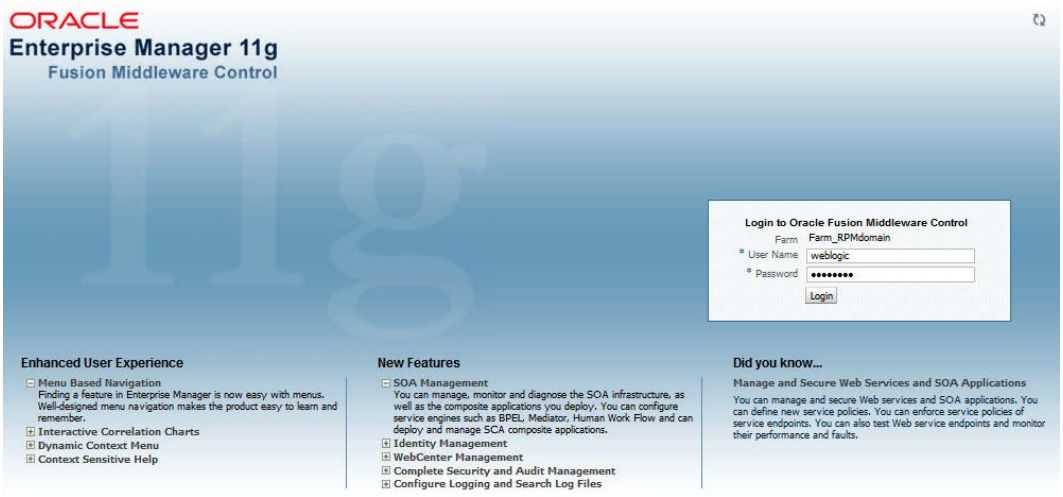


Associate Policy Store to Database

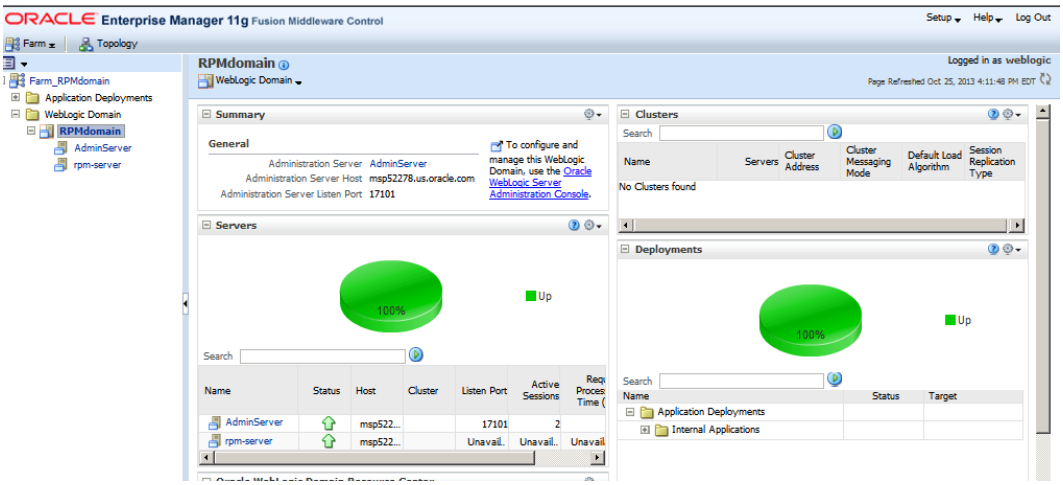
Associate Policy Store to Database

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

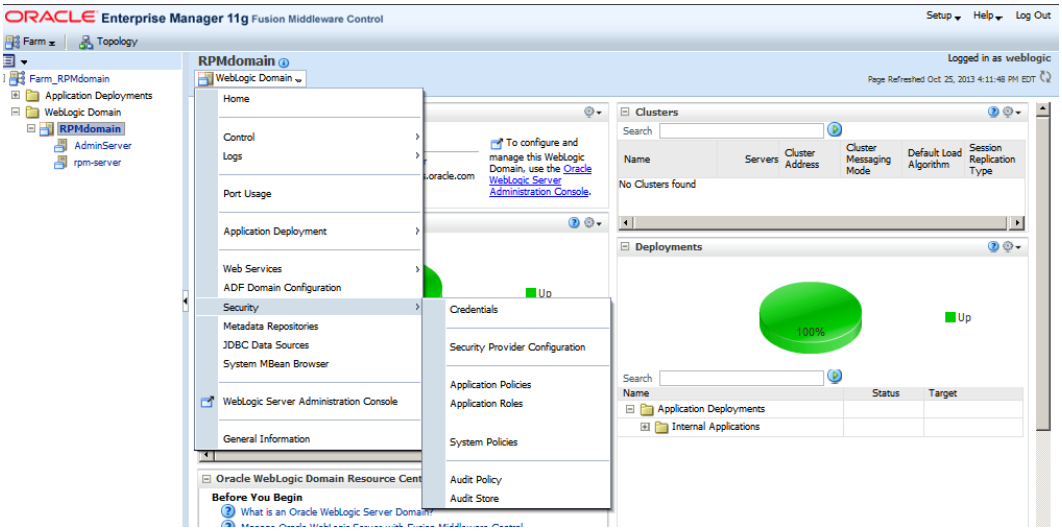
1. Login to the WebLogic EM console (i.e.: http://<Host Name >:17101/em).



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

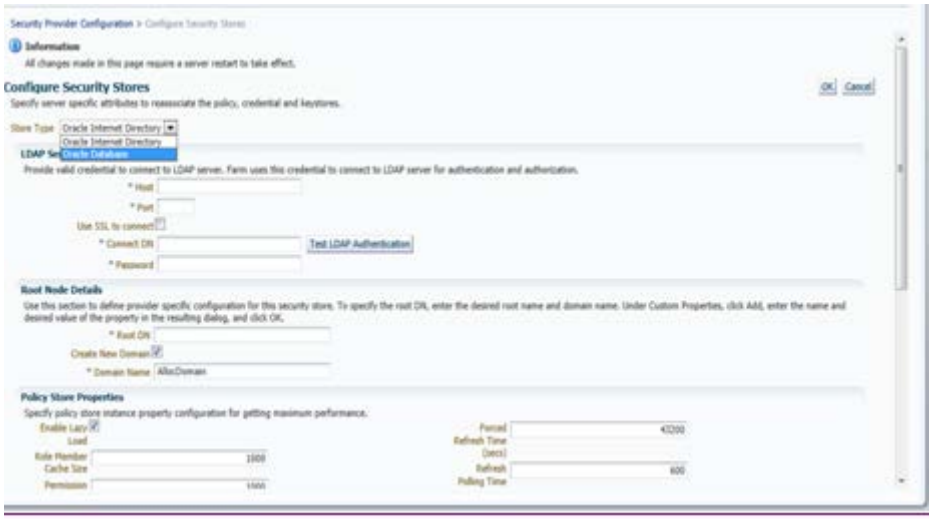


3. Select the dropdown WebLogic Domain->Security->Security Provider Configuration.

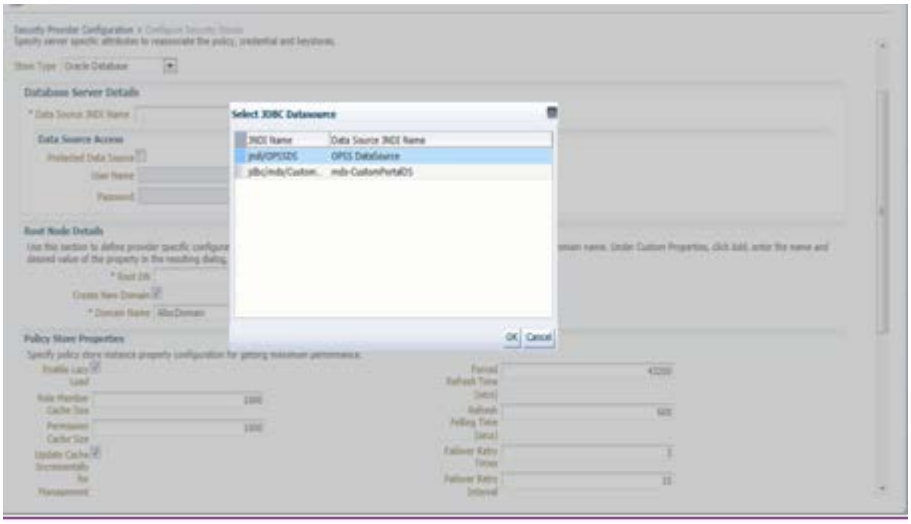


4. Click Change Store Type.

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

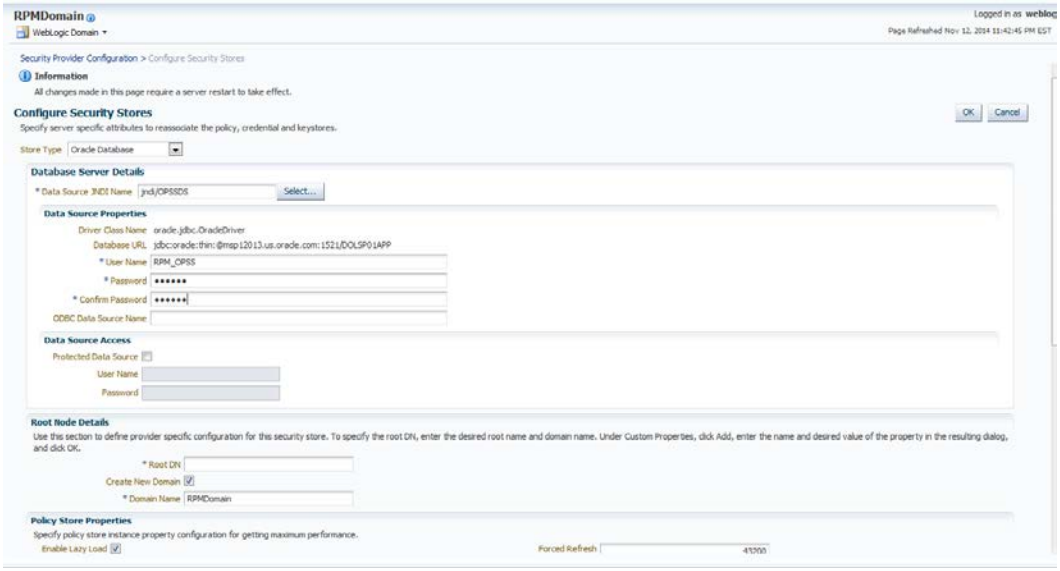


6. Click **Select** and select jndi/OPSSDS JNDI name. Click **OK**.

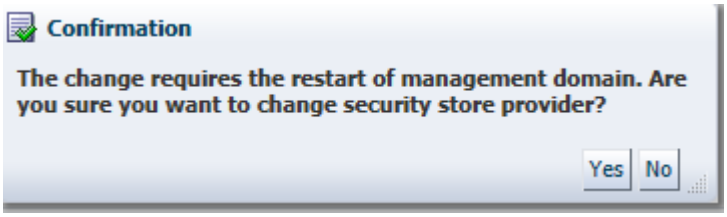


7. Enter the values:
   - Root DN= cn=RPMPolicies
   - Select 'Create New Domain'
   - Domain Name=RPMdomain (This must be the domain name which has been created earlier in this document)

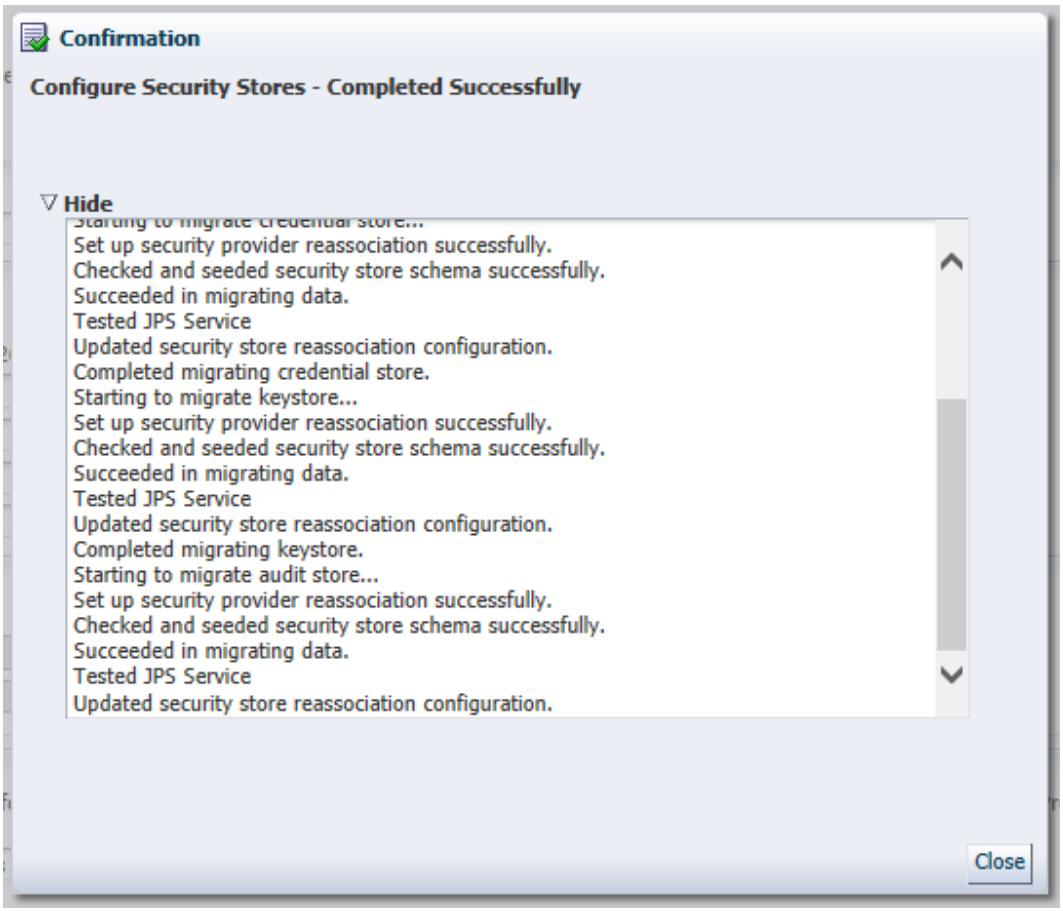
8. Click **OK**.



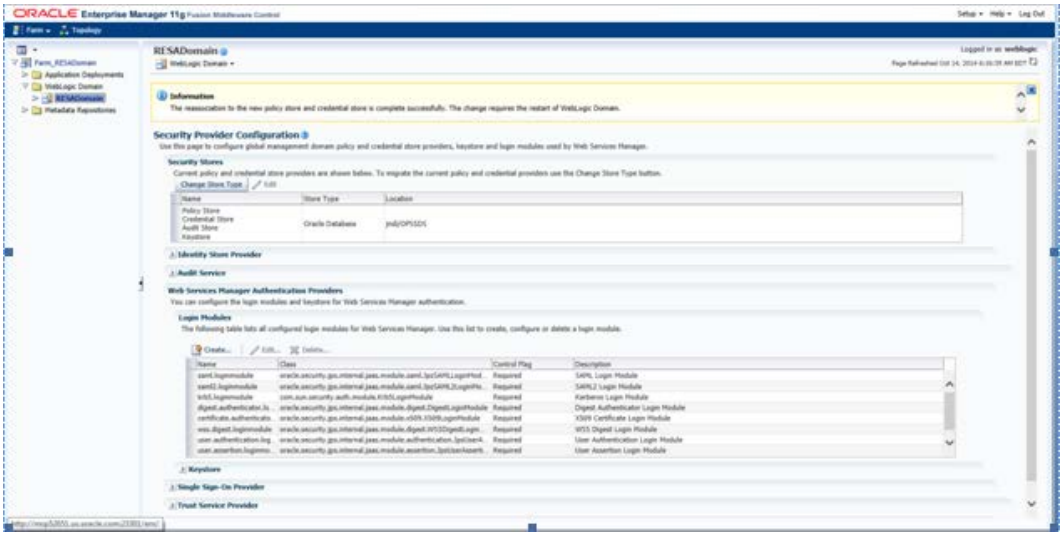
9. Click **Yes**.



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



The following screen appears.

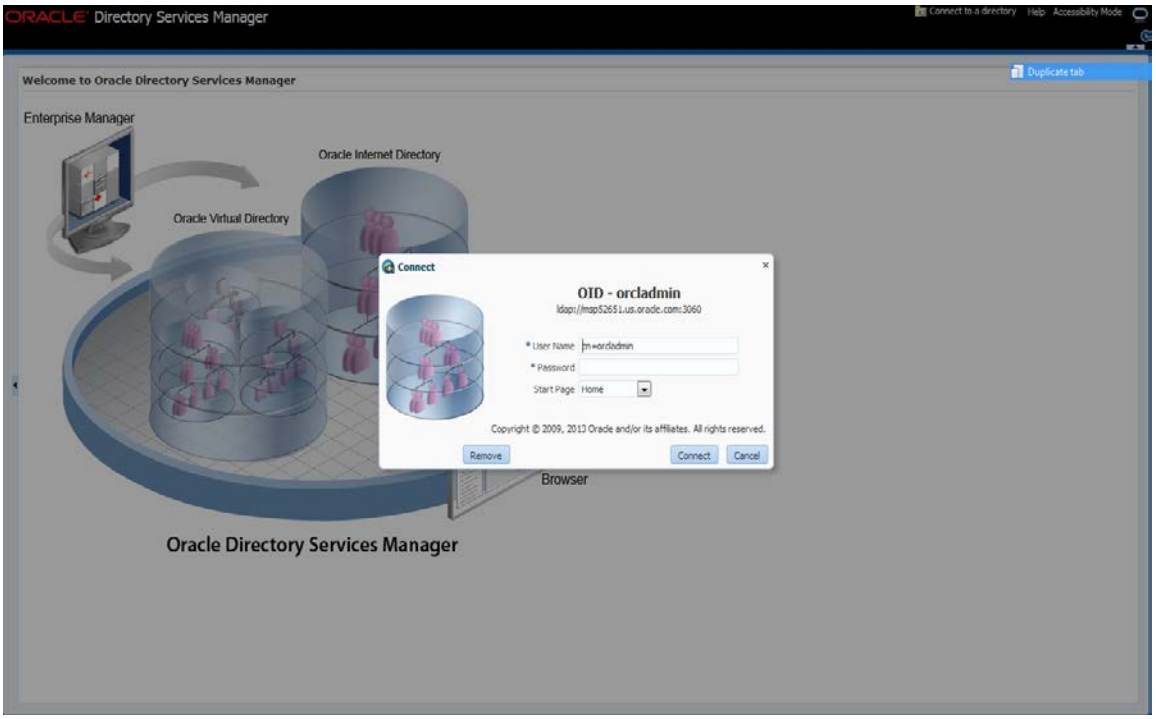


11. Restart the WebLogic domain.
12. Restart the RPMdomain to get the change to take effect.

Configure LDAP authentication Preinstallation Steps (Initial Login to RPM)

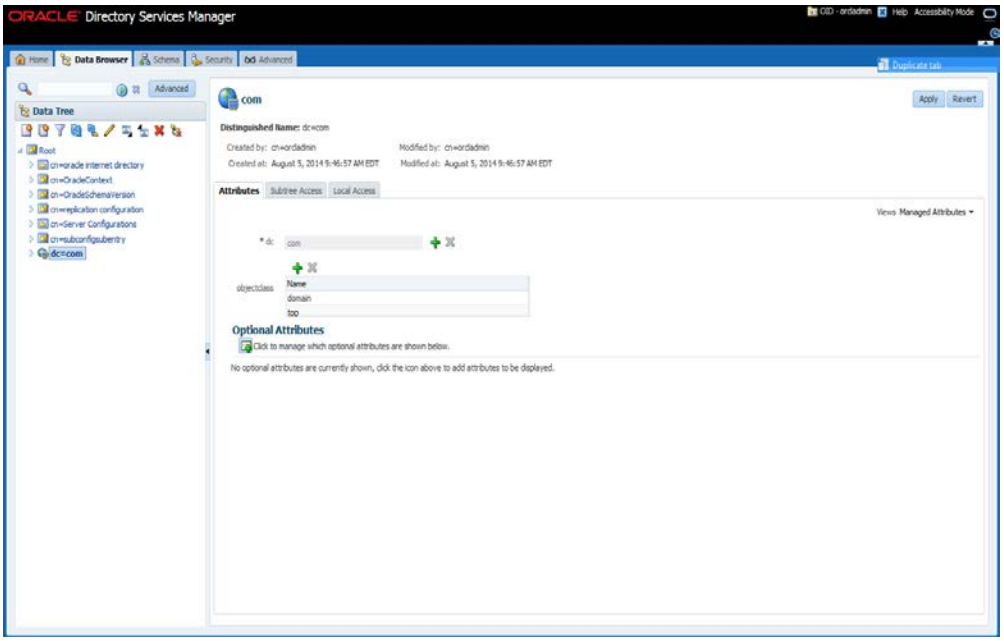
In order to Login to RPM after the installation is done, you need to complete the following pre-installation steps.

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

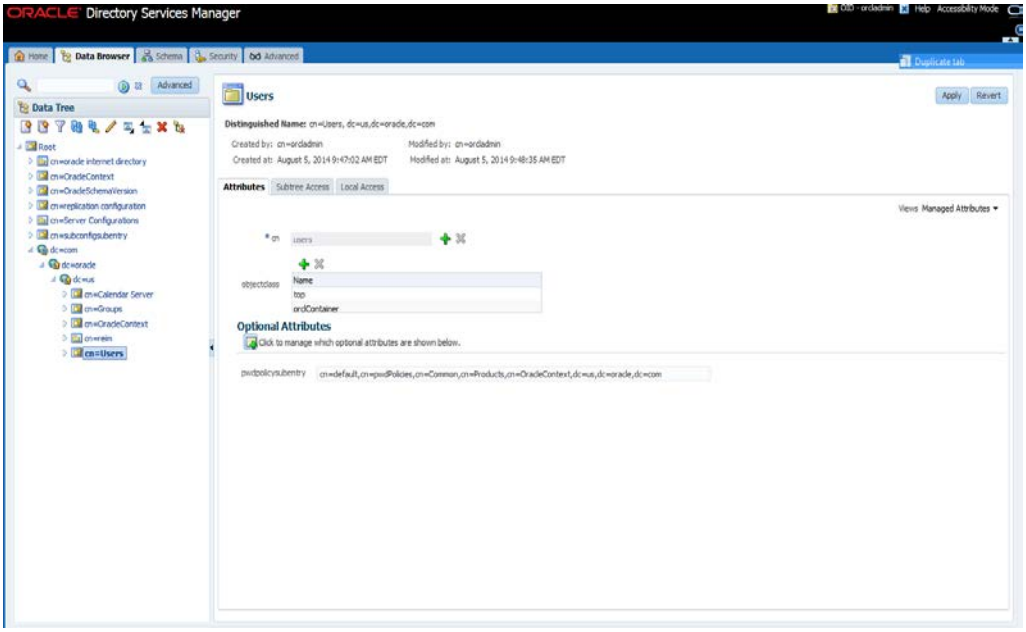


   b. From the OID Connect dialog, click the Connect button.
   c. From the Oracle Internet Directory Welcome Screen, select the Data Browser tab.

   The Data Browser tree shows how to find the "cn=Users" element.



d. From the Data Tree panel of the ODSM screen, navigate to "Users" branch.

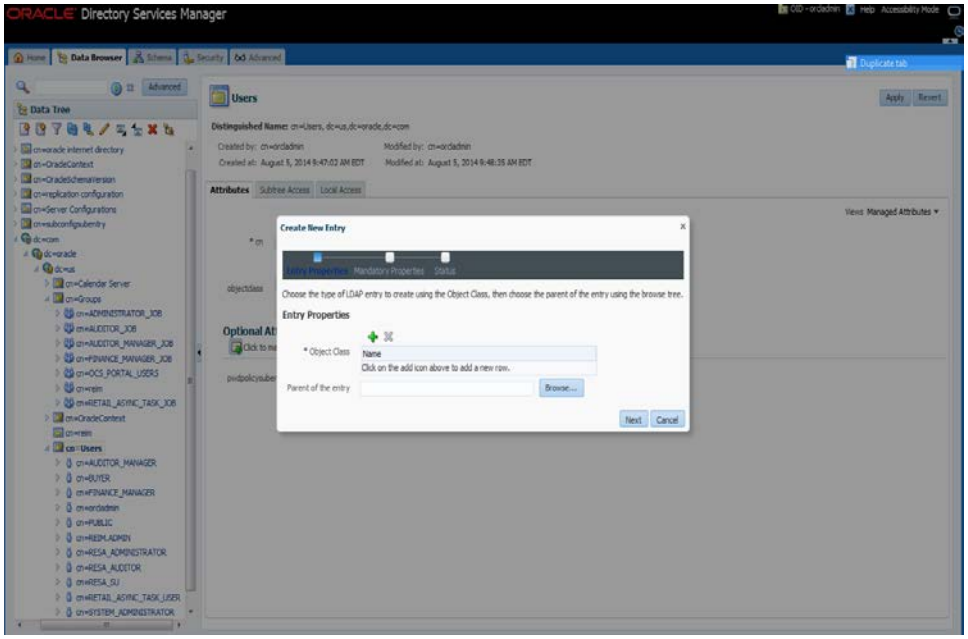


e. On the "Users" screen , press right mouse button with "cn=Users" highlighted and select "Create" from the drop down menu panel

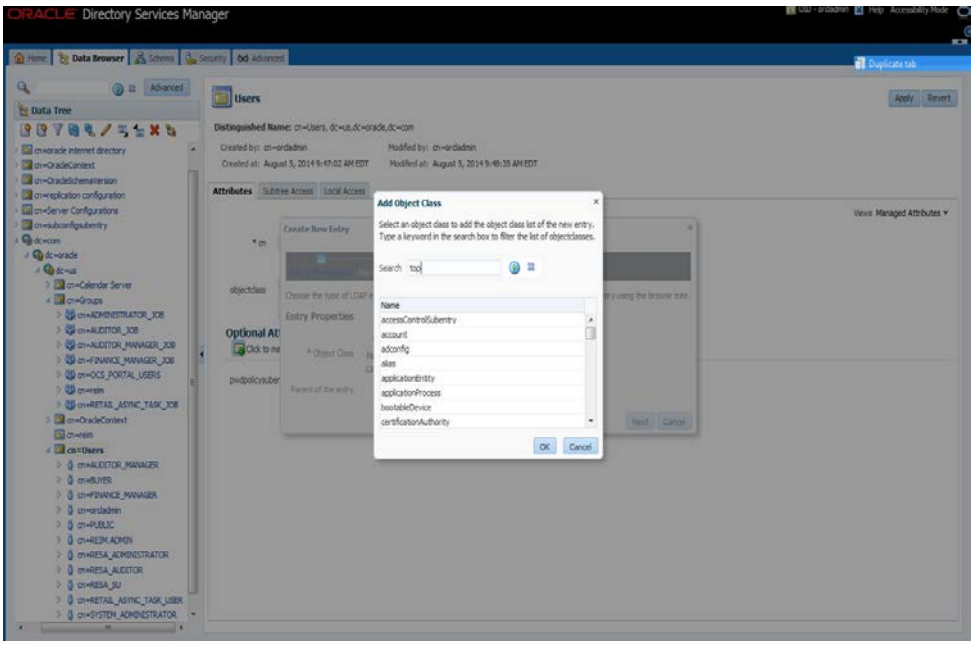
f. In the Object Class field, click the + icon to add the following Object Classes

- top
- orclContainer
- organizationalperson
- orcluser
- person
- orcluserv2
- Inetorgperson

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

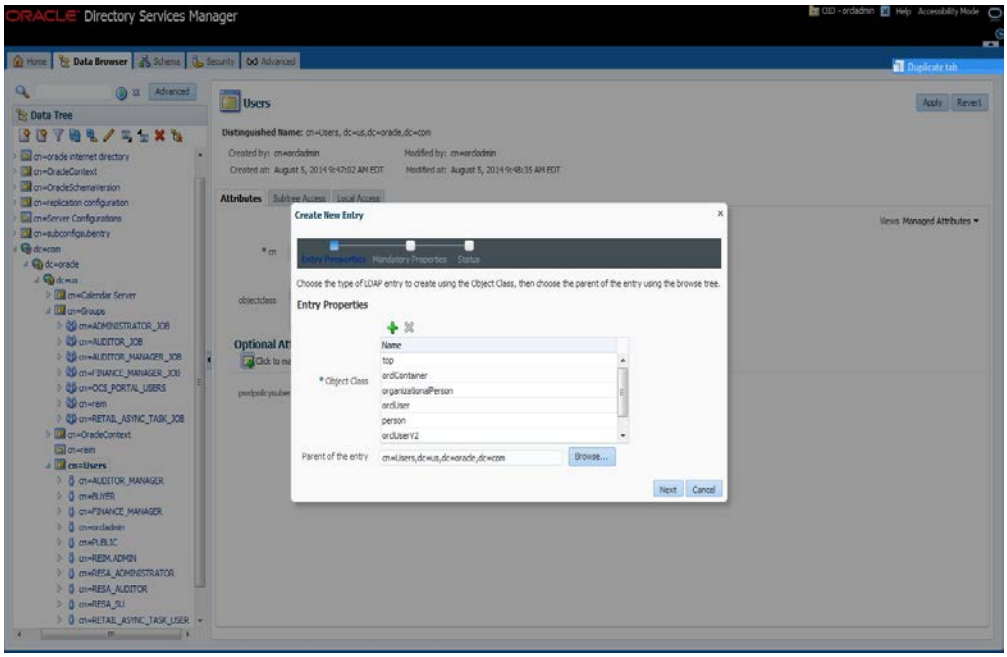


a. From the Add Object Class menu, select the "top" object class.



b. From the Add Object Class menu, select the "orclContainer" object class.

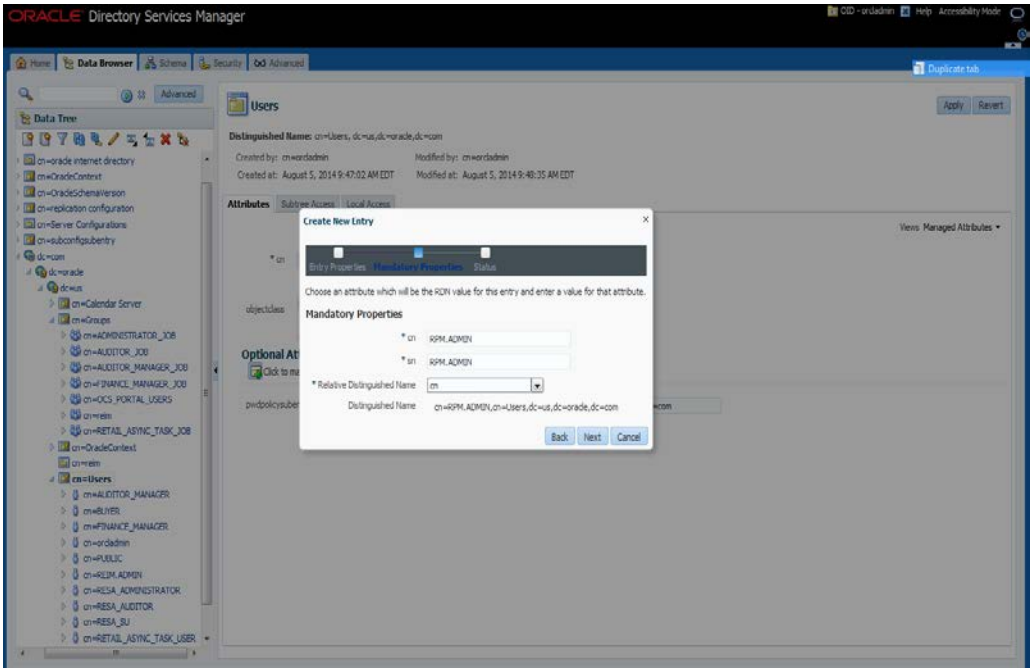
c. When finished adding in all the Object Classes the screen will look as it does below. Then In the Parent of the Entry field enter the following: cn=Users,dc=us,dc=oracle,dc=com



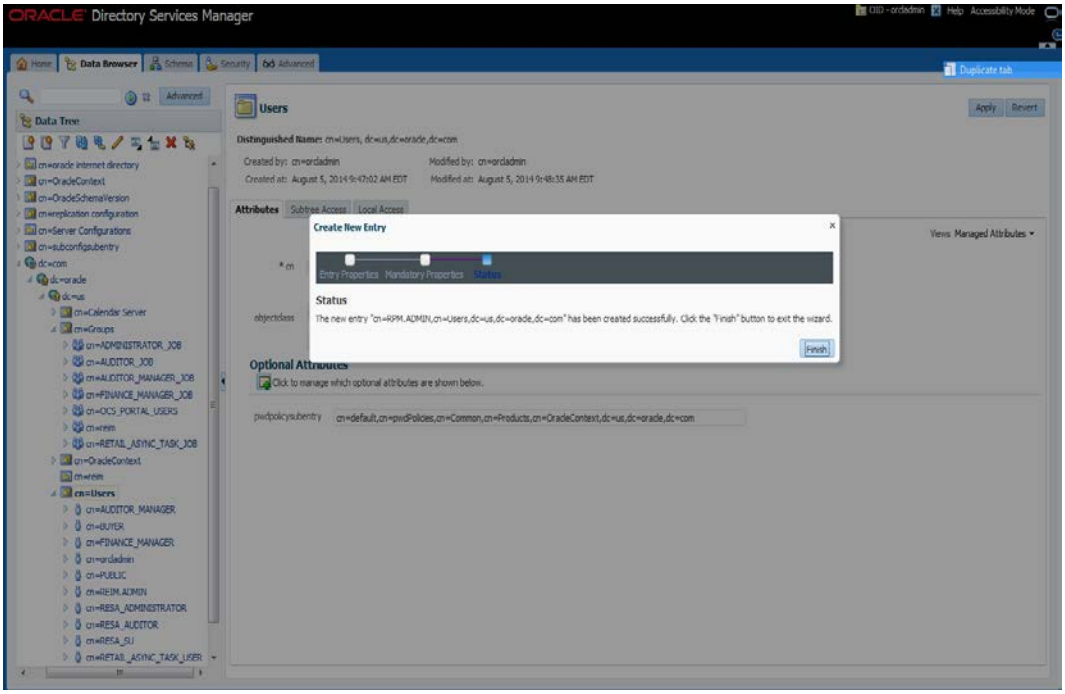
d. Click Next. The Mandatory Properties dialog is displayed.

e. From Mandatory Properties dialog, enter in the following and click next:

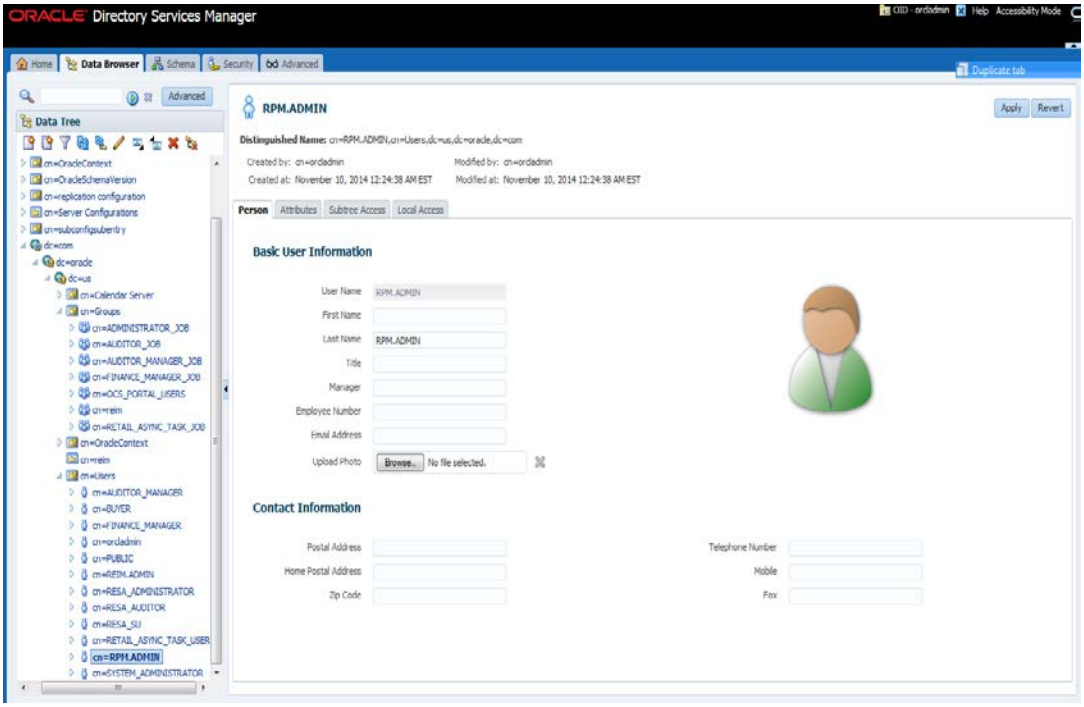
- cn= RPM.ADMIN
- sn= RPM.ADMIN
- Relative Distinguished = cn



f. Make sure the information on screen is correct. Press "Finish" button to create the "RPM.ADMIN" user.



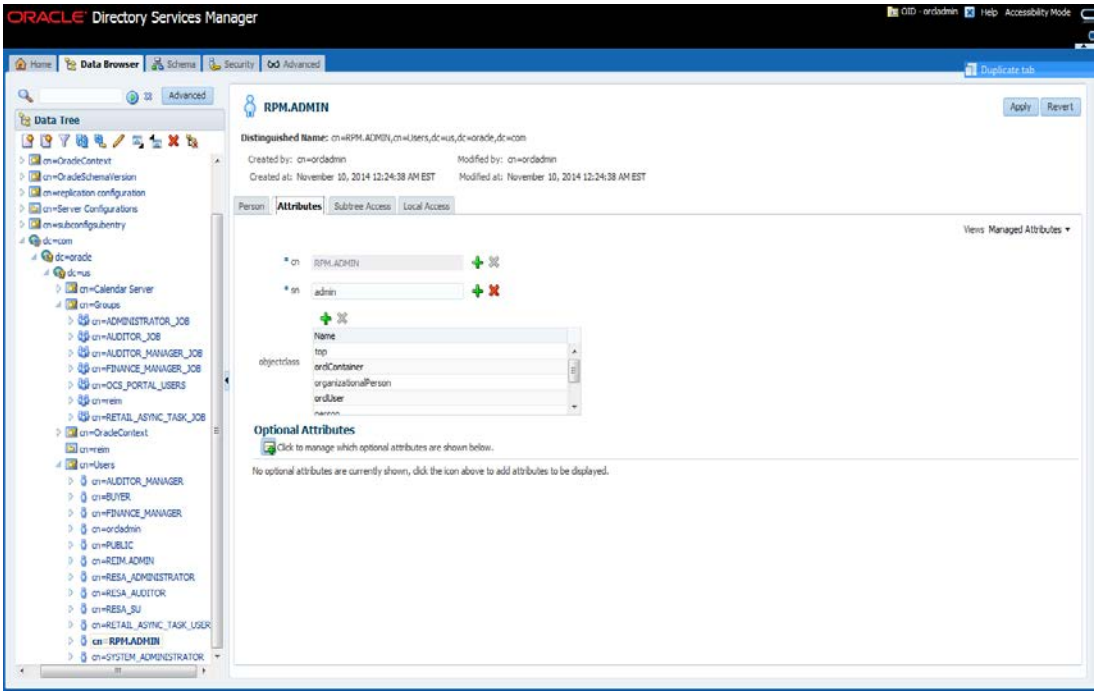
When the "RPM.ADMIN" user is created a screen similar to the one below is displayed when clicking on the new RPM.ADMIN user.



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

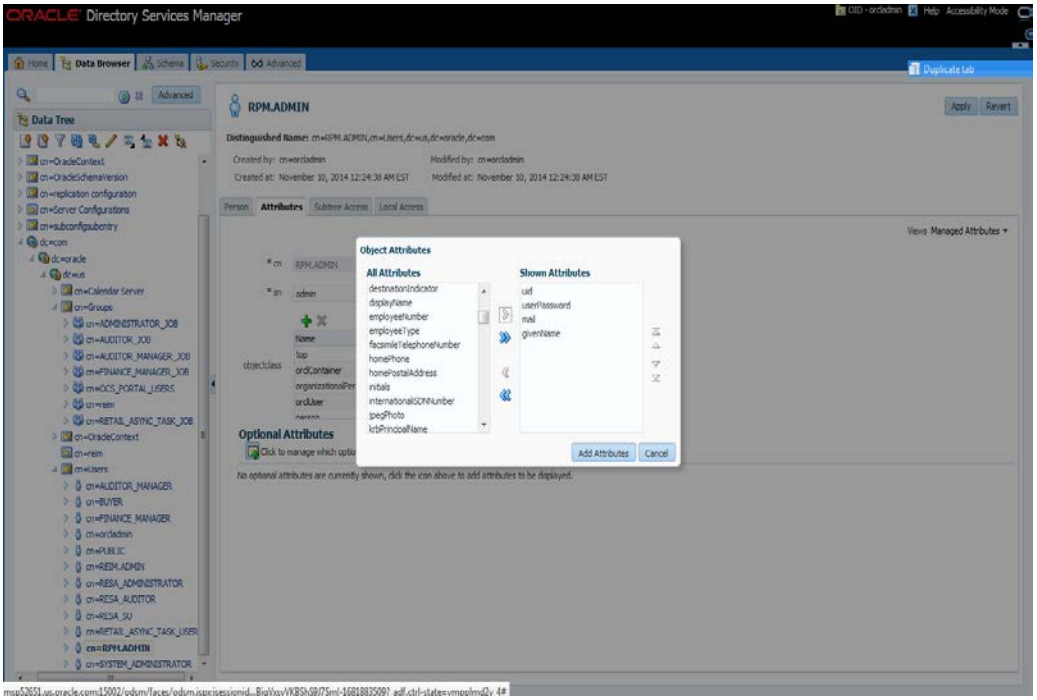
- First Name: <rpm>
- Last Name: <admin>
- Email Address: <rpm.admin@mycompany.com>

h. Click the Attributes tab.



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

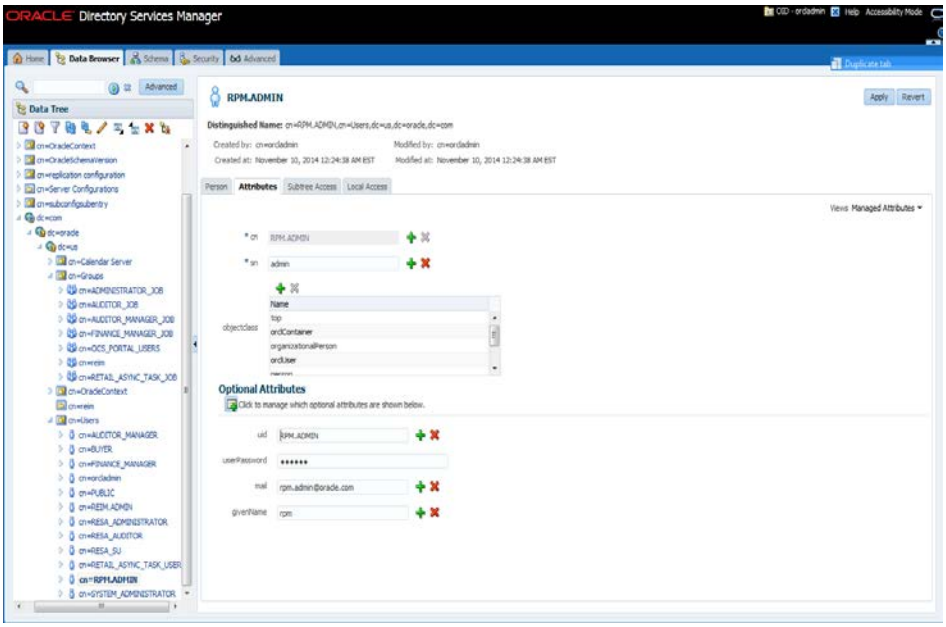
- Given Name: <rpm>
- Mail: <rpm.admin@mycompany.com>
- Uid: RPM.ADMIN
- User Password: <password>

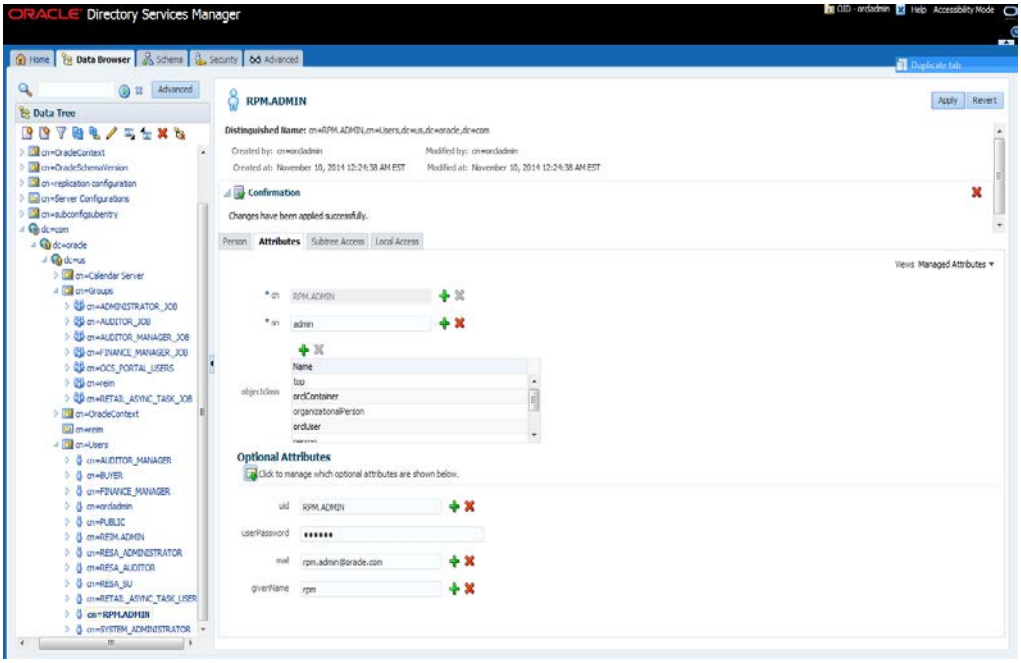


j. Enter the following information and click Apply:

- Given Name: <rpm>

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



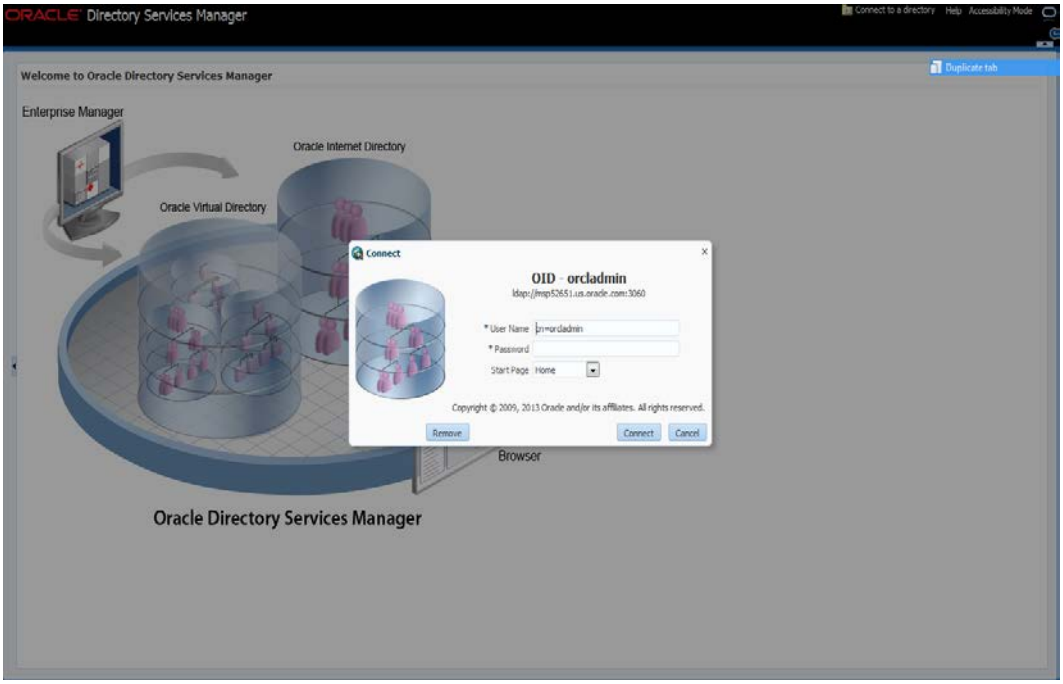


2. Create the Application Admin user who will have access (Login) to RPM.
   a. I f you are installing other Merchandising applications you should have already created RETAIL.USER. If you do not have RETAIL.USER already created in LDAP, create "RETAIL.USER" following the same procedure described for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

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

> **Note:** The RPM code looks for a group named “rpm_secure_users” so it is imperative that the group be named “rpm_secure_users”.

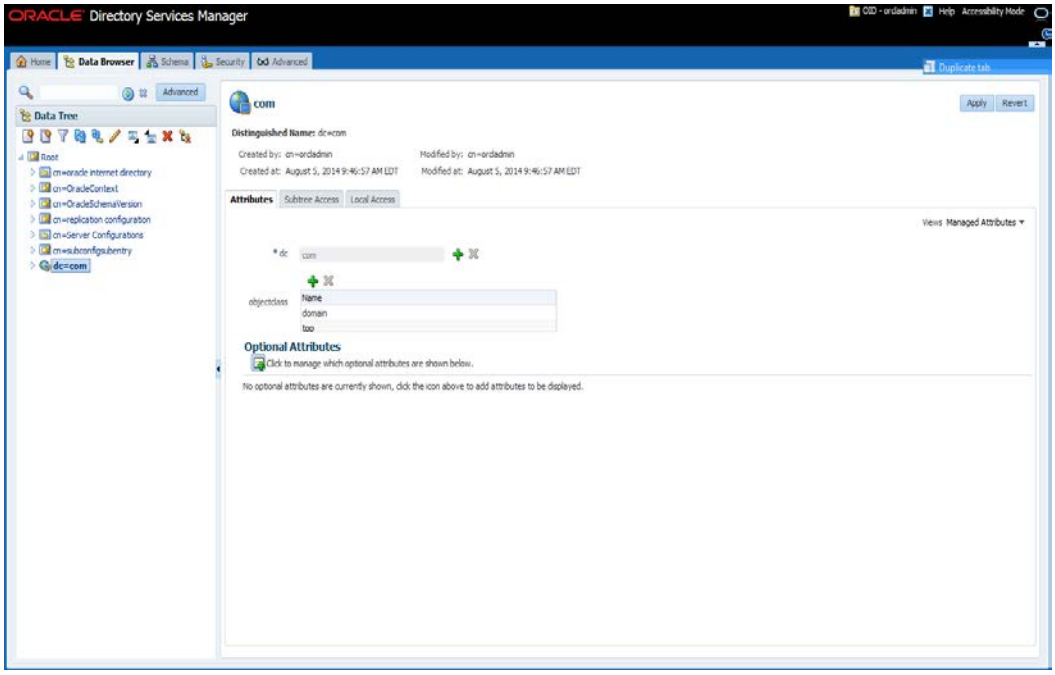
4. Example: Using OID 11.1.1.7, the steps to follow are:
   a. Open your OID connection by launching odsm (Oracle Directory Services Manager). A screen similar to the following is displayed.
   b. Click Connect to a directory and select your OID directory.



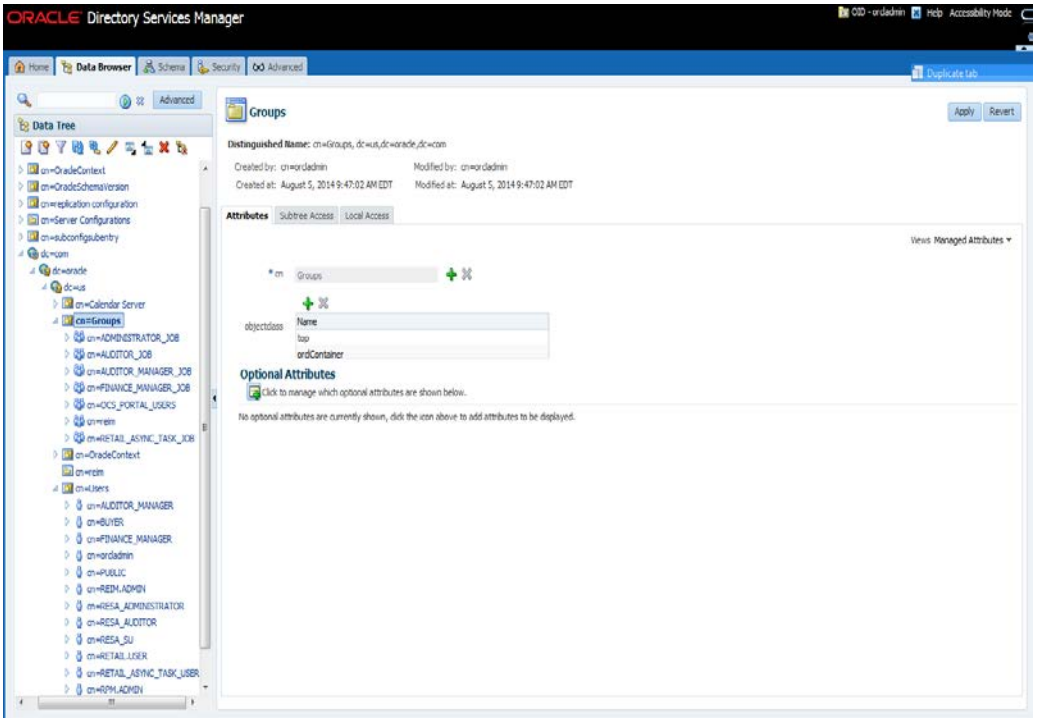
c. From the OID Connect dialog, click the **Connect** button.

d. From the Oracle Internet Directory Welcome Screen, select the Data Browser tab.

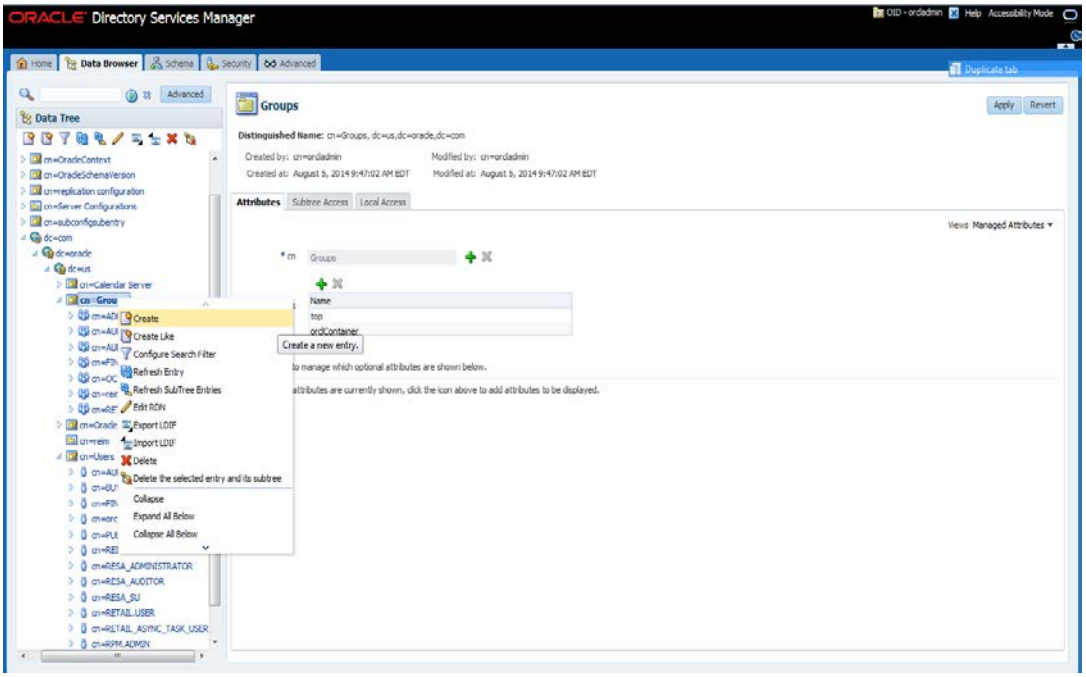
The DataBrowser tree shows how to find the "cn=Group" element.



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

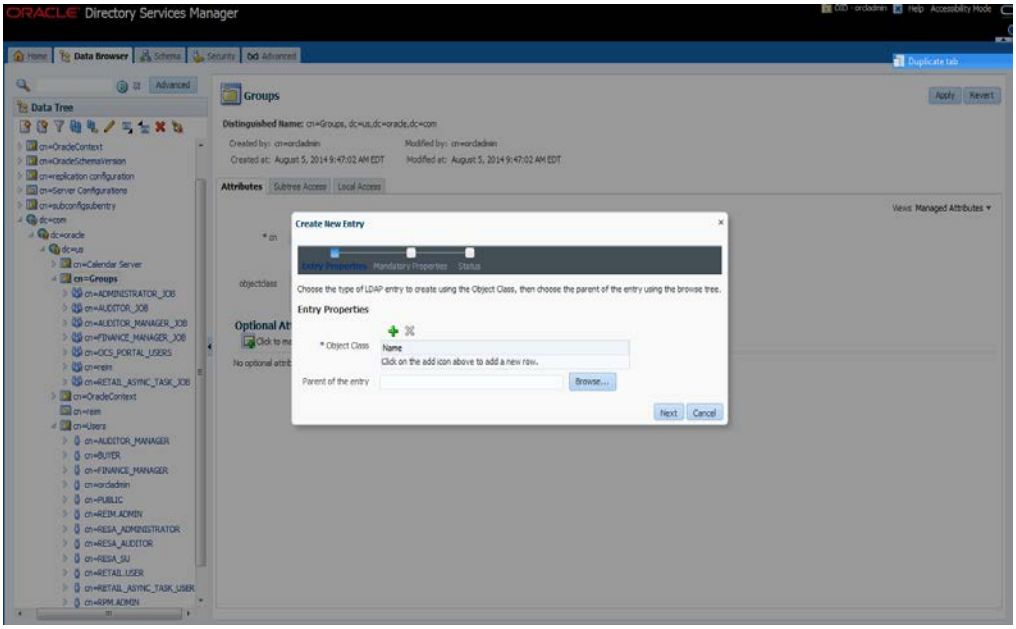


f. Right-click cn=Groups and select **Create**.

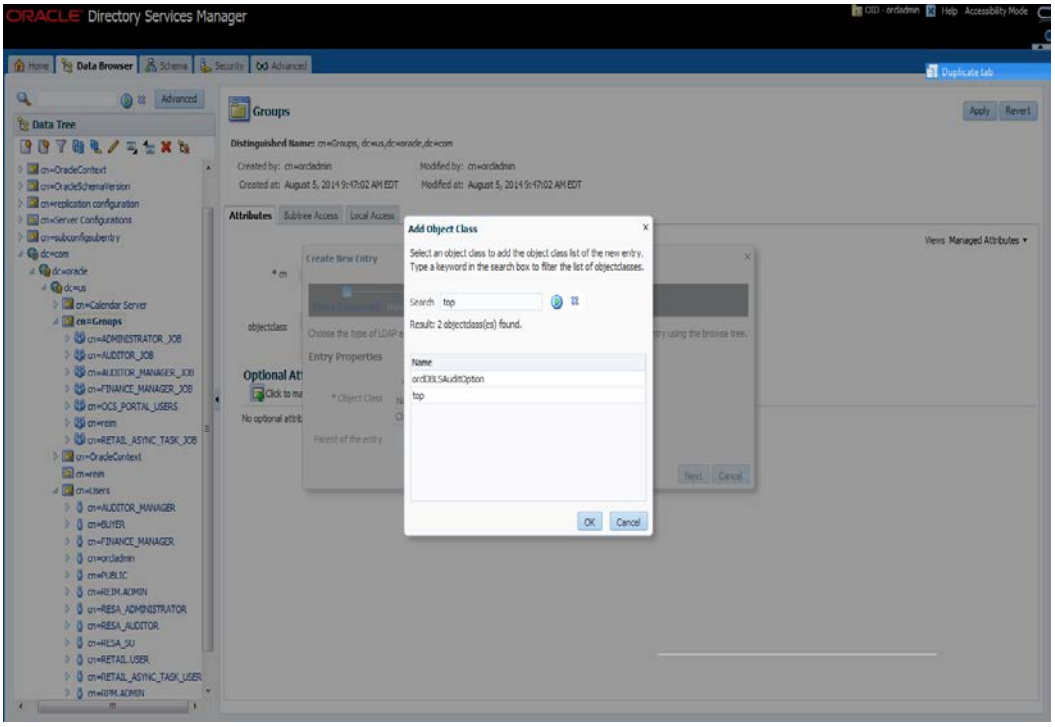


g. From the Create New Entry dialog, click the + icon and add the following Object Classes:

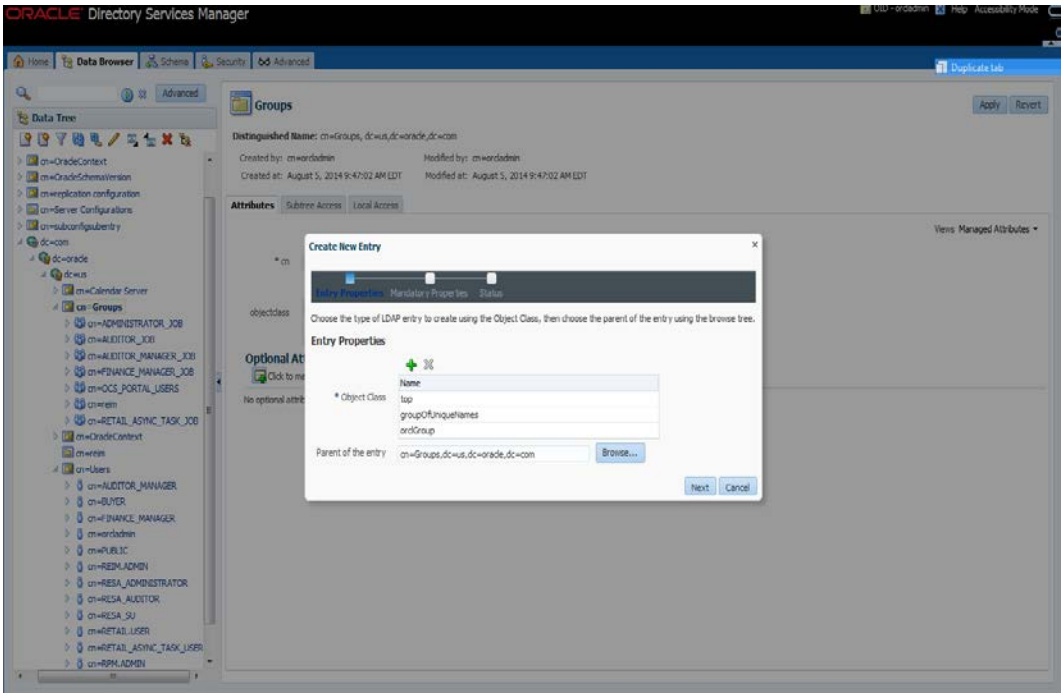
- top
- groupOfUniqueNames
- orclGroup



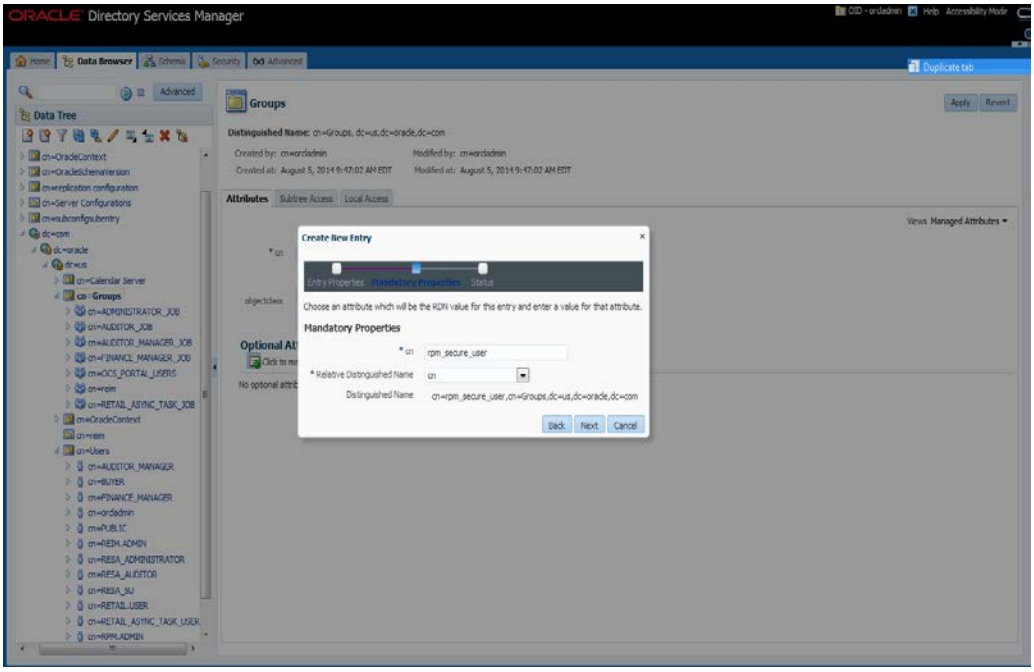
h. From the Add Object Class drop down menu select **top**.



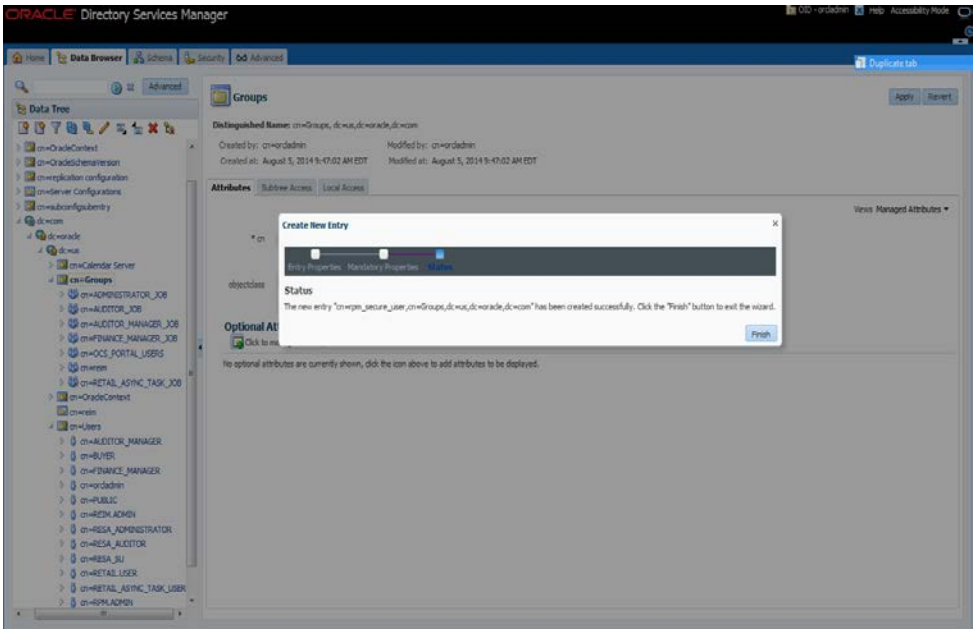
i. When finished adding all the object classes the screen will look like as follows: Enter: cn=Groups,dc=us,dc=oracle,dc=com into the Parent of the entry field. Click **Next**



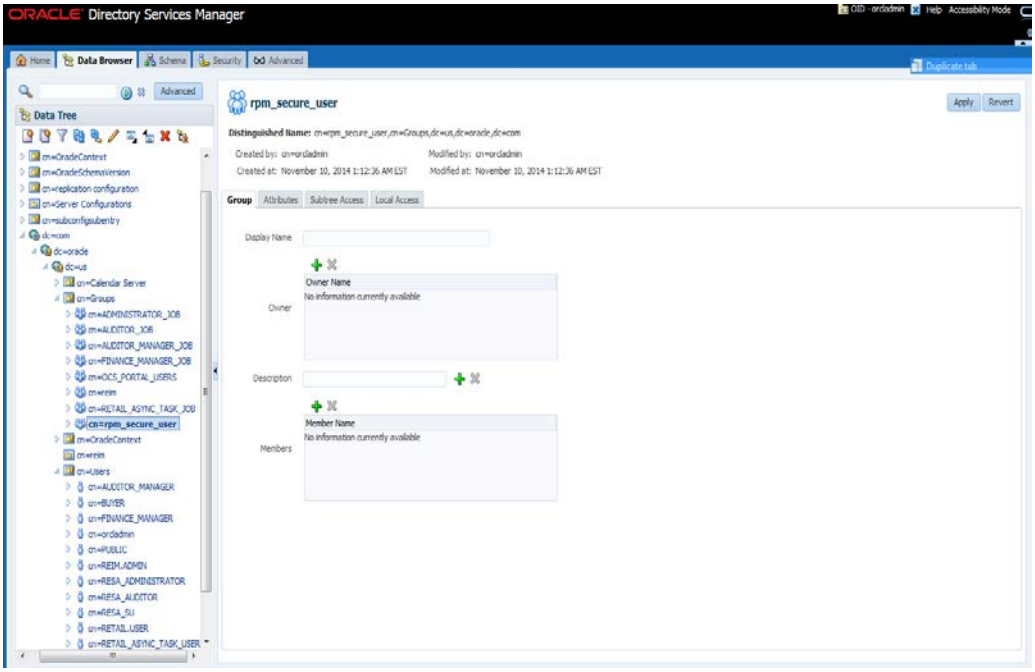
j. On the "*cn" text field enter: "rpm_secure_users". On Relative Distinguished Name field enter: cn and click **Next**.



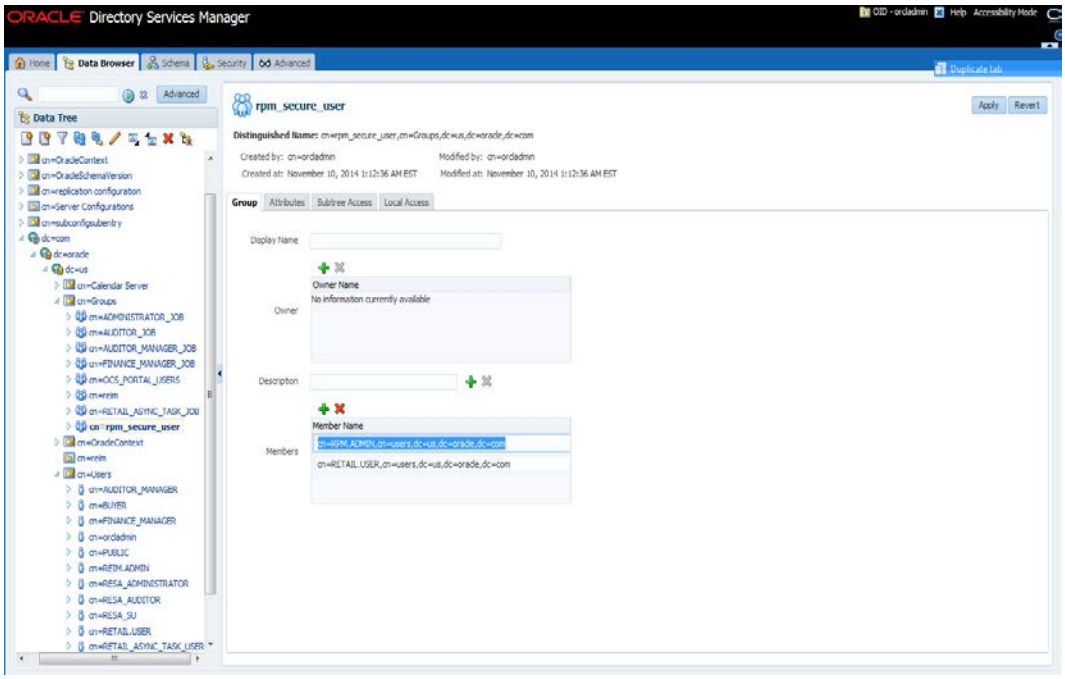
k. Click **Finish**.



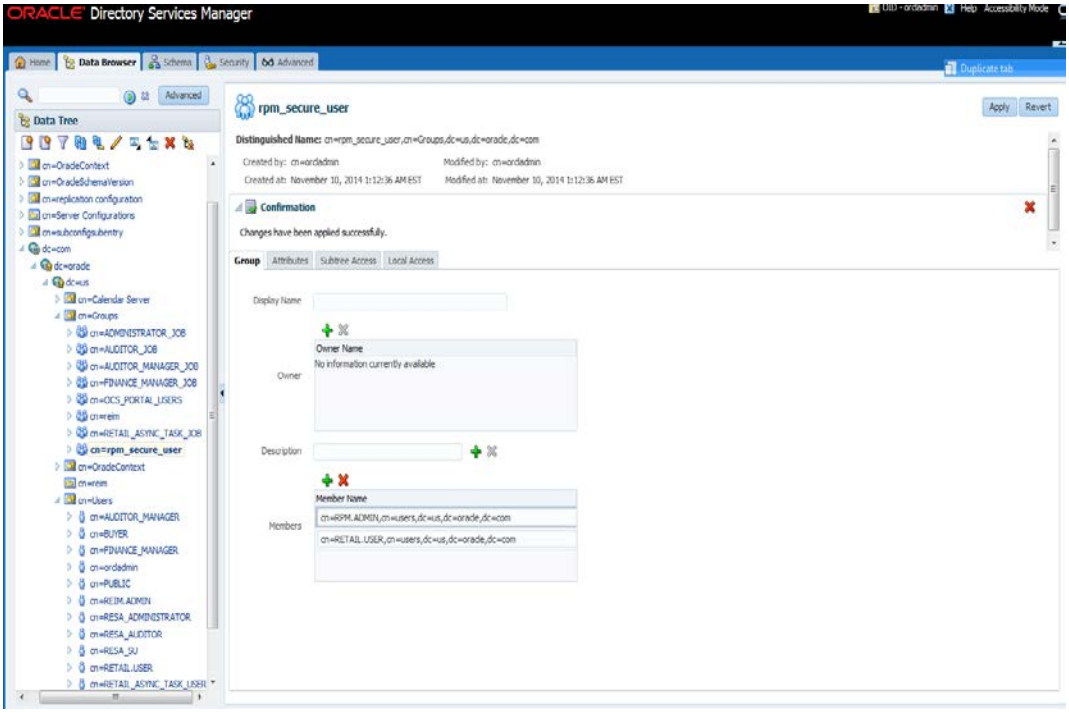
After applying the changes, your screen should look similar to the following when clicking on the rpm_secure_users group:



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

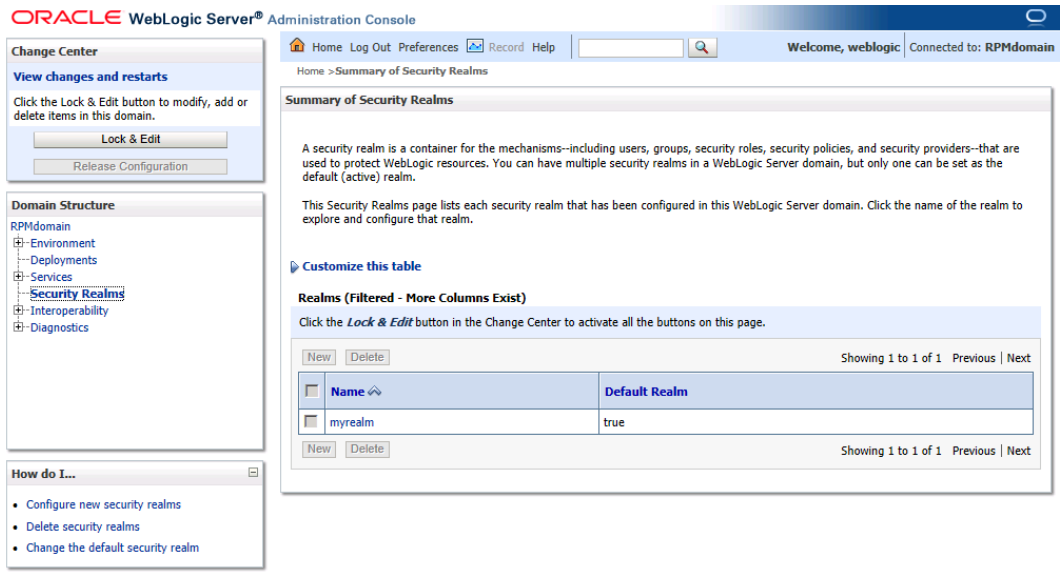


m. Click Apply. You will get the screen below.

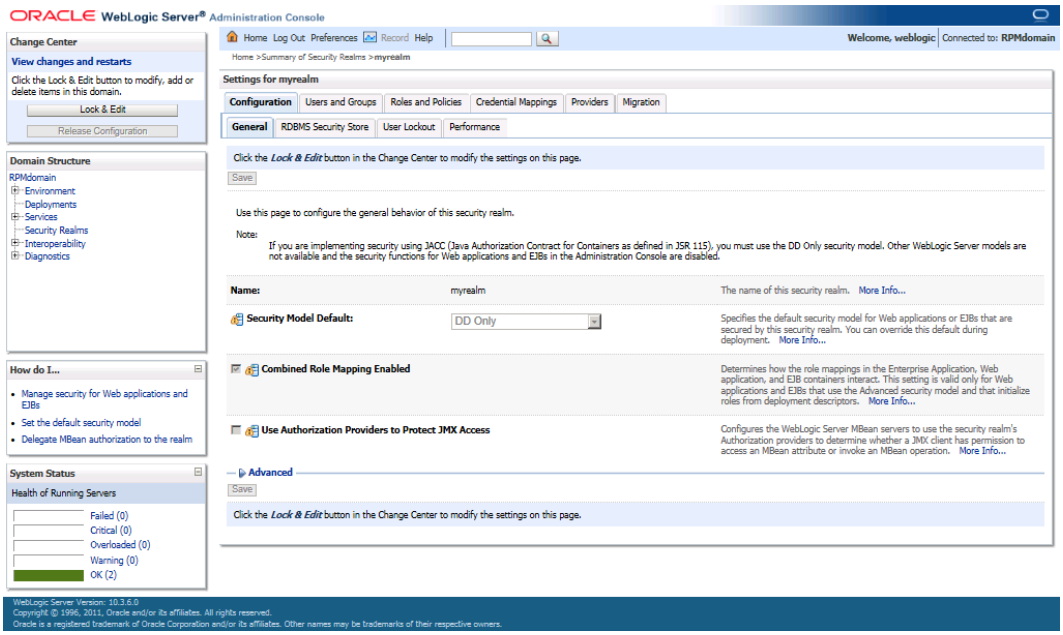


Create OID Authentication Provider

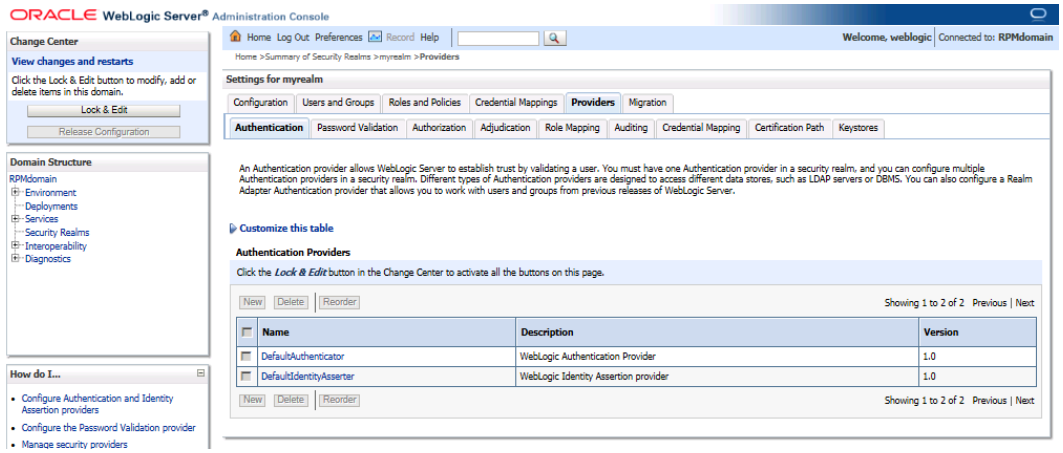
1. Log in to the Administration Console.
   http://<host>:<port>/console/
2. In the Domain Structure frame, click **Security Realms**.



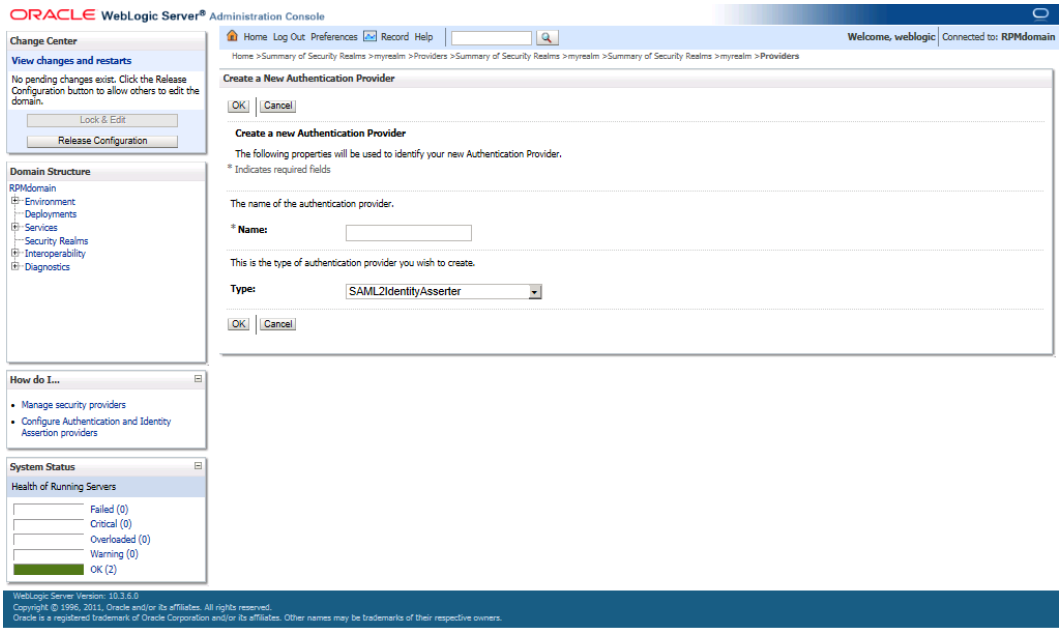
3. In the Realms table, click **myrealm**. The Settings for myrealm page is displayed.



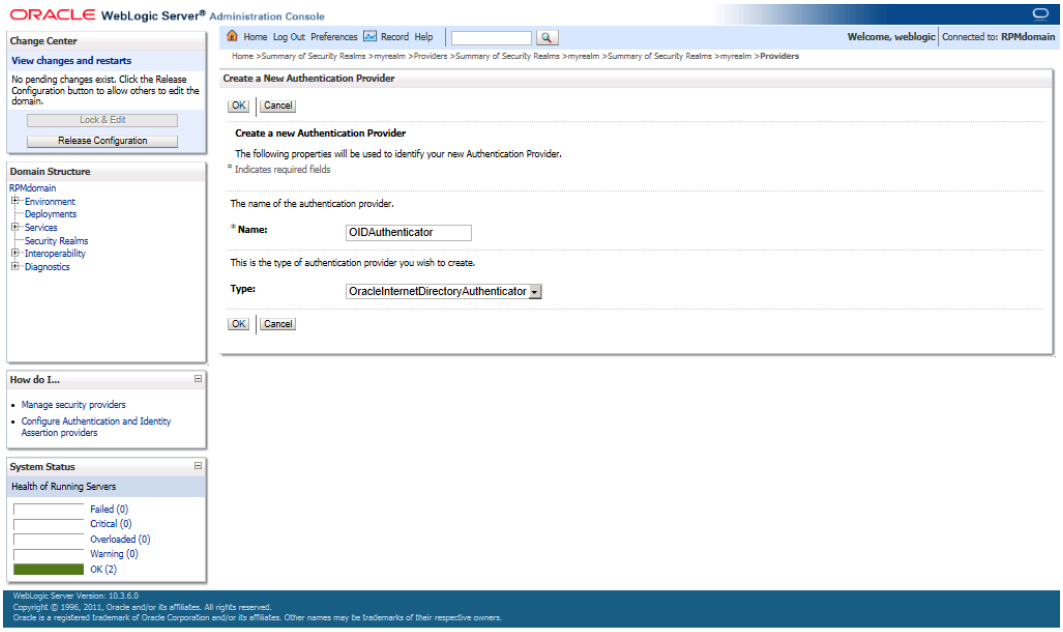
4. Click the Providers tab.



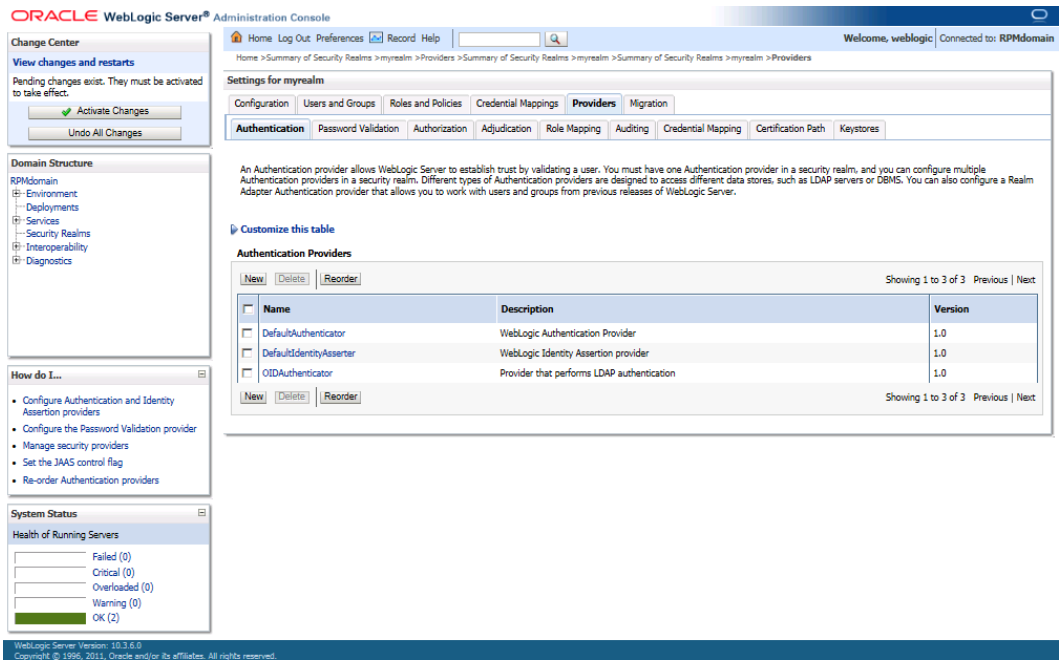
5. Click **Lock & Edit** and then click **New**. The Create a New Authentication Provider page is displayed.



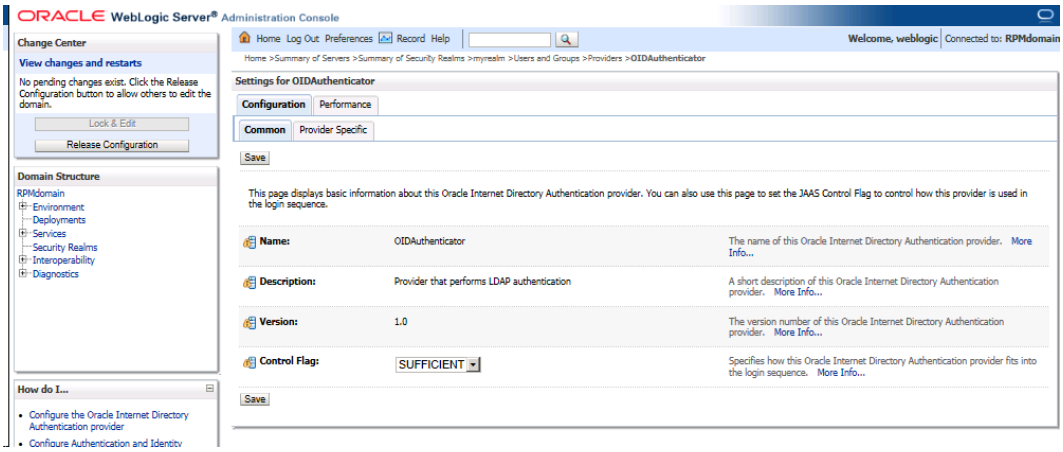
6. Enter OIDAuthenticator in the Name field and select OracleInternetDirectoryAuthenticator as the type.



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



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

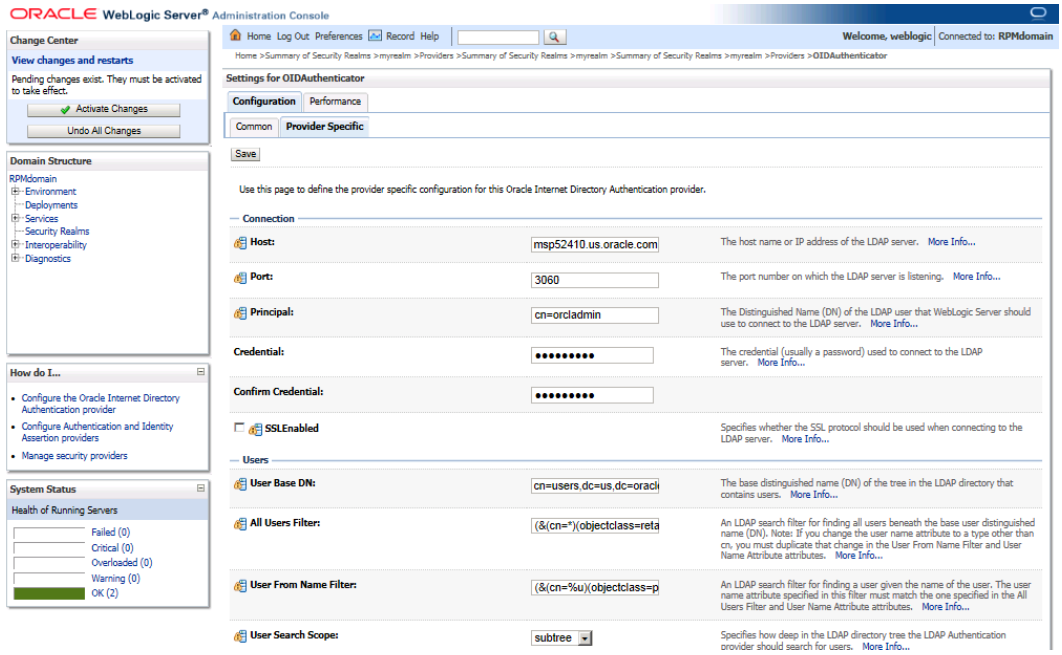


9. Set the Control Flag field to SUFFICIENT and click **Save**.
10. Click the Provider Specific tab.
11. Supply your LDAP connection and credentials.

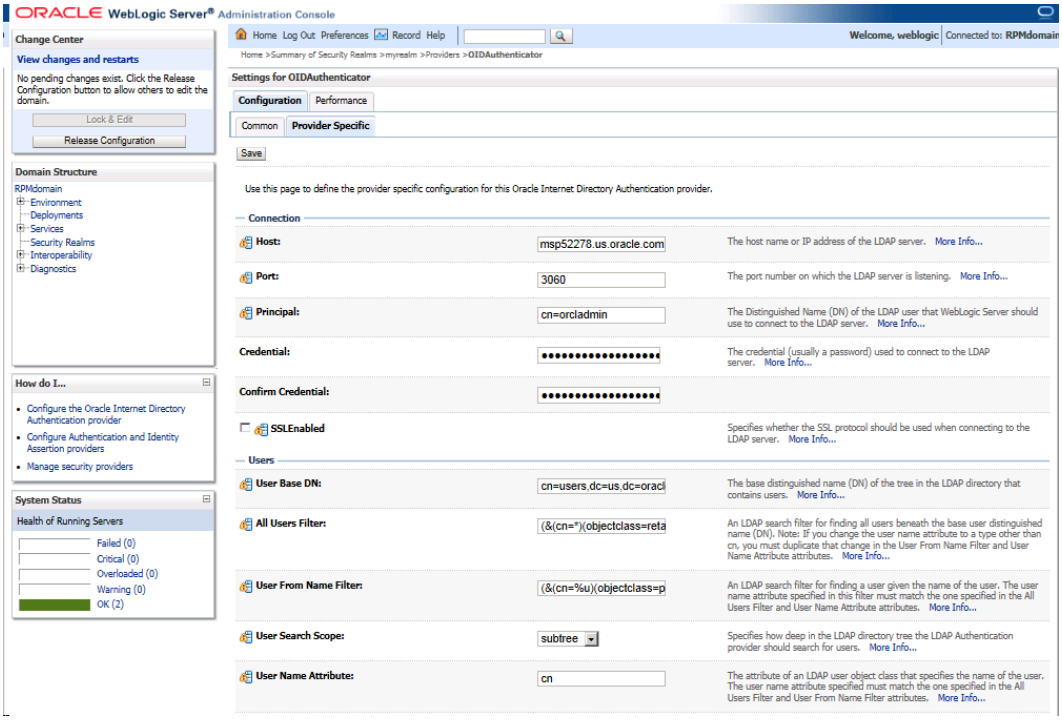
    The entries below are examples only. You should match the entries to your OID

    - Host: <OID Server name> (Example: msp12068.us.oracle.com)
    - Port: <OID port> (Example: 3060 or 389)
    - Principal: <cn=orcladmin> (provide the OID admin user)
    - Credential: <password> (provide the password of cn=orcladmin)
    - User Base DN: (Example: cn=Users,dc=us,dc=oracle,dc=com)
    - Group Base DN: (Example: cn=Groups,dc=us,dc=oracle,dc=com)

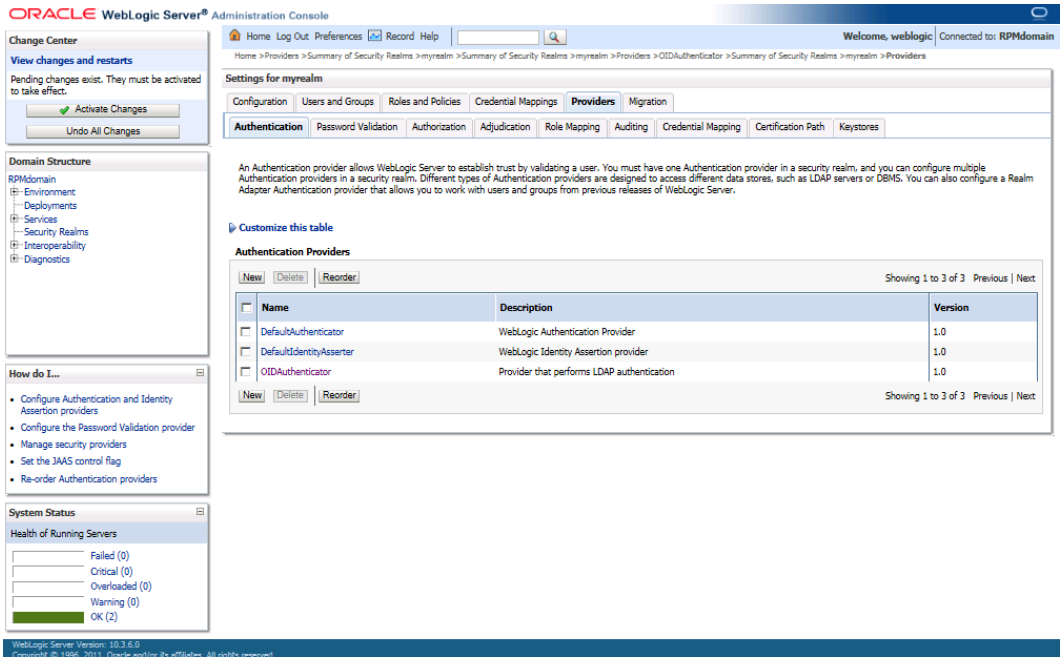
- All Users Filter : (&(cn=*)(objectclass=retailUser))



12. Click **Save**.

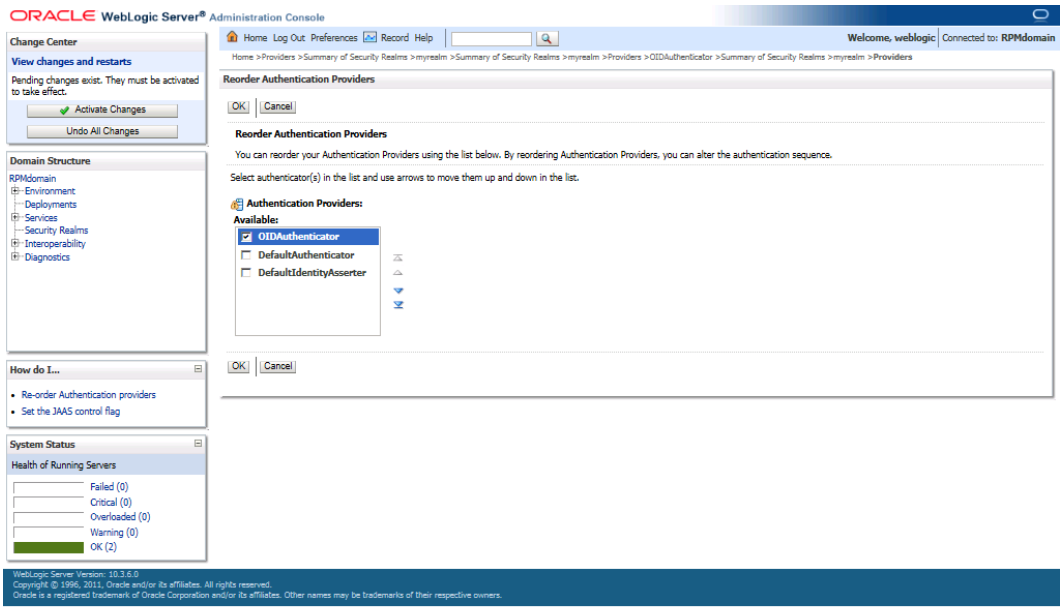


13. Navigate to Security Realms – myrealm – and then click the Providers tab.

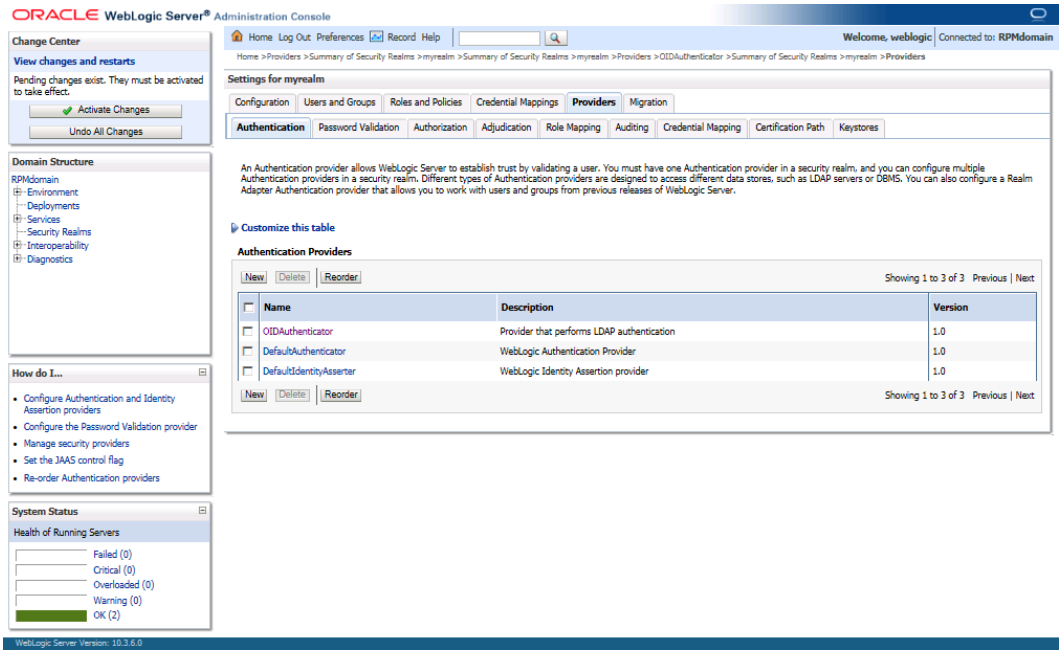


14. Click **Reorder**.

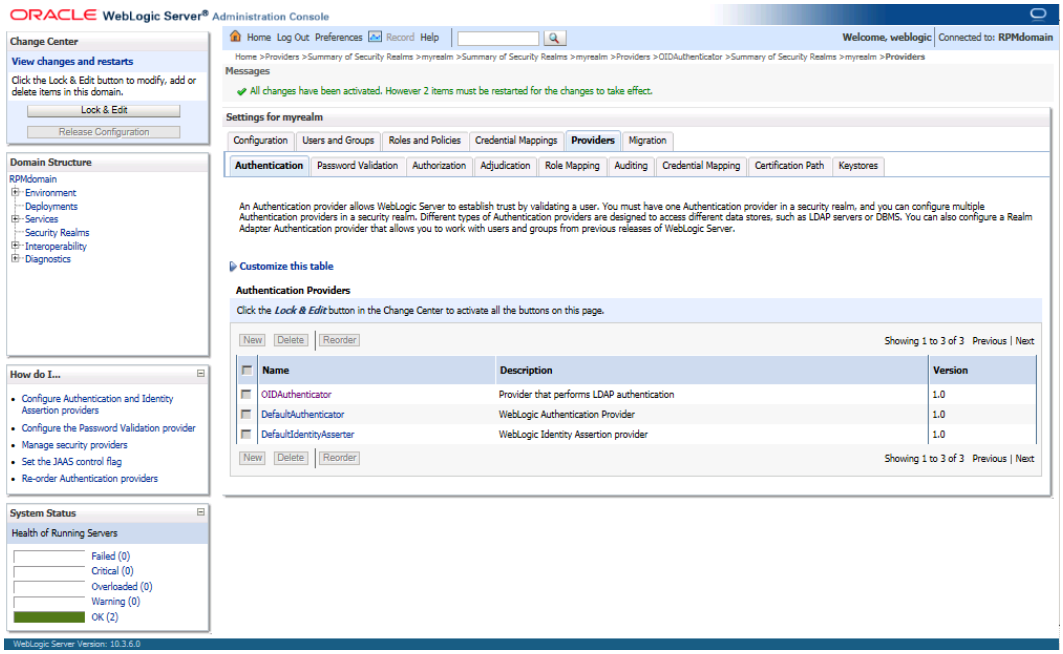
15. Order OIDAuthenticator first and DefaultAuthenticator second.



16. Click **Ok**.



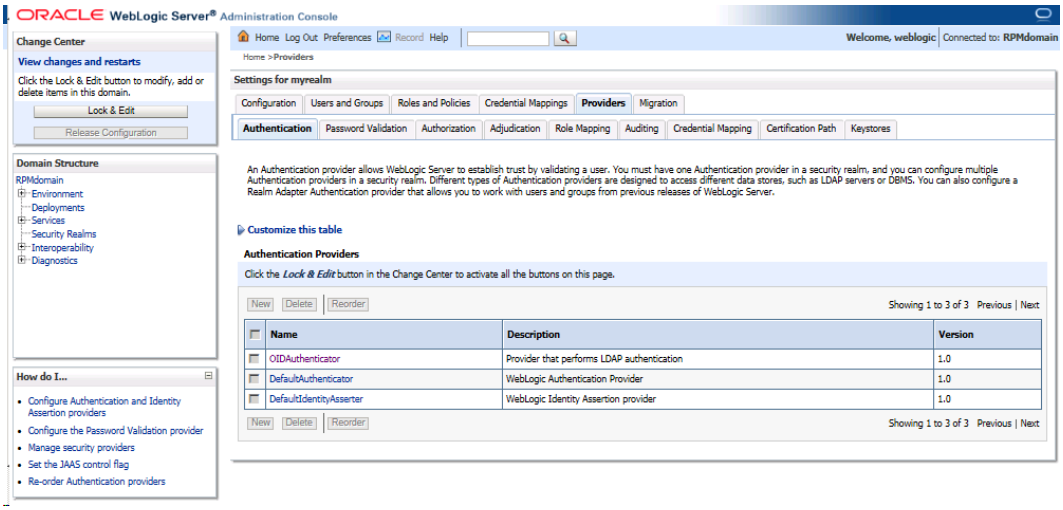
17. Once your changes are saved, click **Activate Changes**.



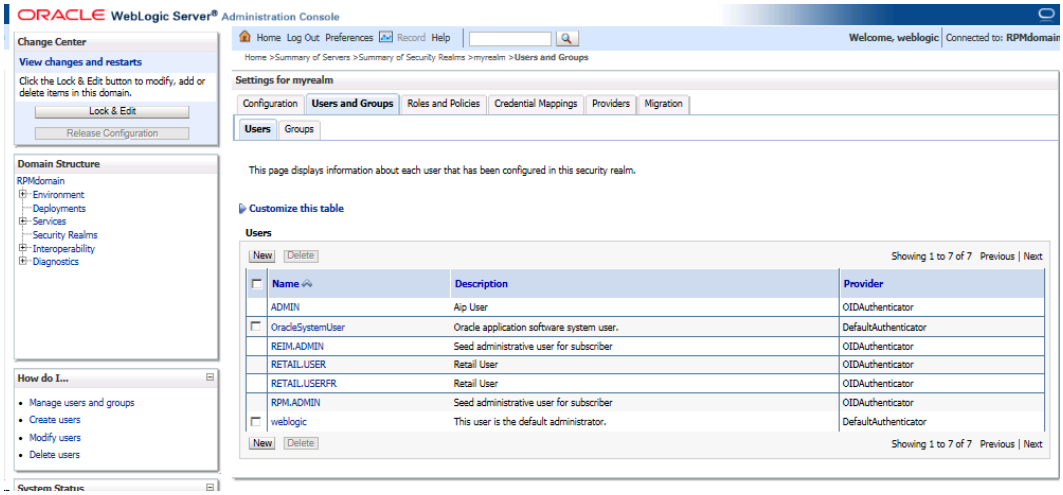
18. Shut down all servers and restart the admin server.

Verify and Set OID Authenticator

1. Log in to the Administration Console.
   http://<host>:<port>/console/
2. In the Domain Structure frame, click Security Realms.
3. In the Realms table, click Default Realm Name. The Settings page is displayed.
4. Click the Providers tab. You must see the OID Provider in that list.



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



Expand the RPM Application Distribution

To expand the RPM application distribution, do the following.

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

   Example: /u00/webadmin/media/rpm

   This location is referred to as STAGING_DIR for the remainder of this chapter.
2. Copy rpm14application.zip to STAGING_DIR and extract its contents.

Clustered Installations – Preinstallation Steps

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

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

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

   input.taskqueue.name = taskQueue

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

Provide the Hibernate4 Jar File

The RPM application requires hibernate4 jar files to be installed. These files should be downloaded from http://www.hibernate.org.
Extract the required Hibernate 4.1.0 jar files and place them within the application servers STAGING_DIR/rpm/application/hibernate4 directory before running the installer. The installer will then install the jar files within the application for you.

The required jars are as follows:

- hibernate-commons-annotations-4.0.*.jar
- hibernate-core-4.1.*.jar
- hibernate-ehcache-4.1.*.jar
- hibernate-jpa-2.0-api-1.0.*.jar
- jboss-logging-3.1.*.jar
- jboss-transaction-api_1.1_spec-1.0.*.jar

The required jar files are located in the <HIBERNATE_EXTRACT_DIR>\hibernate-release-4.1.0.Final.tgz\hibernate-release-4.1.0.Final.tar\hibernate-release-4.1.0.Final\lib\required\directory.

Run the RPM Application Installer

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

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

1. Change directories to STAGING_DIR/rpm/application.
2. Set the ORACLE_HOME, WEBLOGIC_DOMAIN_HOME and JAVA_HOME environment variables. ORACLE_HOME should point to your WebLogic installation. JAVA_HOME should point to the Java 7.0 (1.7.0) JDK . WEBLOGIC_DOMAIN_HOME should point to your WebLogic domain.
3. If a secured datasource is going to be configured you also need to set "ANT_OPTS" so the installer can access the key and trust store that is used for the datasource security:

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

An example of this would be:

```
export ANT_OPTS="-
Djavax.net.ssl.keyStore=/u00/webadmin/product/wls_retail
/wlserver_10.3/server/lib/msp52278.keystore -Djavax.net.ssl.keyStoreType=jks -
Djavax.net.ssl.keyStorePassword=retail123 -Djavax.net.ssl.trustStore=/
u00/webadmin/product/wls_retail
/wlserver_10.3/server/lib/msp2278.keystore -Djavax.net.ssl.trustStoreType=jks
-Djavax.net.ssl.trustStorePassword=retail123"
```

4. If you are using an X server such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
5. Run the install.sh script. This launches the installer. After installation is complete, a detailed installation log file is created (rpm14install.<timestamp>.log).

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

Resolving Errors Encountered During Application Installation

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

See Appendix: Common Installation Errors in this document for some common installation errors.

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

Clustered Installations – Post-Installation Steps

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

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

Review and/or Configure Oracle Single Sign-On

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

Create the RPM SSO provider in the RPMdomain:

1. Shut down all the servers of the Weblogic Domain created.
2. Once you copy the contents to <INSTALL_DIR> copy the rpm14-security.zip present in <INSTALL_DIR>/ rpm/application/rpm14 to the WEBLOGIC_DOMAIN_HOME/lib and extract it contents in the folder.
3. Start the domain admin server.
4. Log into the WebLogic console.
5. Navigate to: security realms -> myrealm (default realm) -> providers.
6. Start a Lock and Edit session.
7. Click New provider.
8. Select the provider type from the list: RpmWlsSsoAuthenticator.
9. Set the provider name (Default: RpmSsoAuthenticator).
10. Click **Ok**.
11. Open the new provider configuration.
12. Under Common, set the Control Flag to SUFFICIENT.
13. Click **Provider Specific**.
14. Check that the GroupName is set to the name of the group used for RPM secure users (rpm_secure_users by default).
15. All other values under the Provider Specific tab can be left as the default value.
16. Click **Ok**.
17. On the provider list, click **Reorder**.
18. Move the RpmWlsSsoAuthenticator to the top of the list, or above the DefaultAuthenticator.
19. Click **Ok**.
20. Click **Activate Changes**.
21. Shutdown the domain.
22. Update the <WLS_HOME>/wlserver_10.3/server/lib/weblogic.policy file to include the <WEBLOGIC_DOMAIN_HOME>/lib directory:

```
grant codeBase "file:<WEBLOGIC_DOMAIN_HOME>/lib/-"  {
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/RPMDomain/lib/-"
{
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";
};
```

23. Start the admin and managed servers for the domain.

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

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

For example, in mod_wl_ohs.conf set:

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

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

Resource URL: / rpm-client/launch

Protection Level: Protected

Authentication Policy: Protected Resource Policy

Authorization Policy: Protected Resource Policy

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

Protection Level: Unprotected

Authentication Policy: Public Resource Policy

Authorization Policy: Public Resource Policy

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/<38o5n1 >/war/lib. To sign the rpm_client_config.jar file using your alias and keystore, run the jarsigner utility.

Example: jarsigner $WEBLOGIC_DOMAIN_HOME/servers/rpm-managedserver/tmp/_WL_user/rpm/38o5n1 /war/lib/rpm_client_config.jar foo

If you are clustering the application server you need to copy the signed rpm_client_config.jar file to the same path under $ORACLE_HOME on all remote nodes.

Consult the **jarsigner** documentation from Sun for further information on the JAR signing process.

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

Backups Created by Installer

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

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

Test the RPM Application

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

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

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

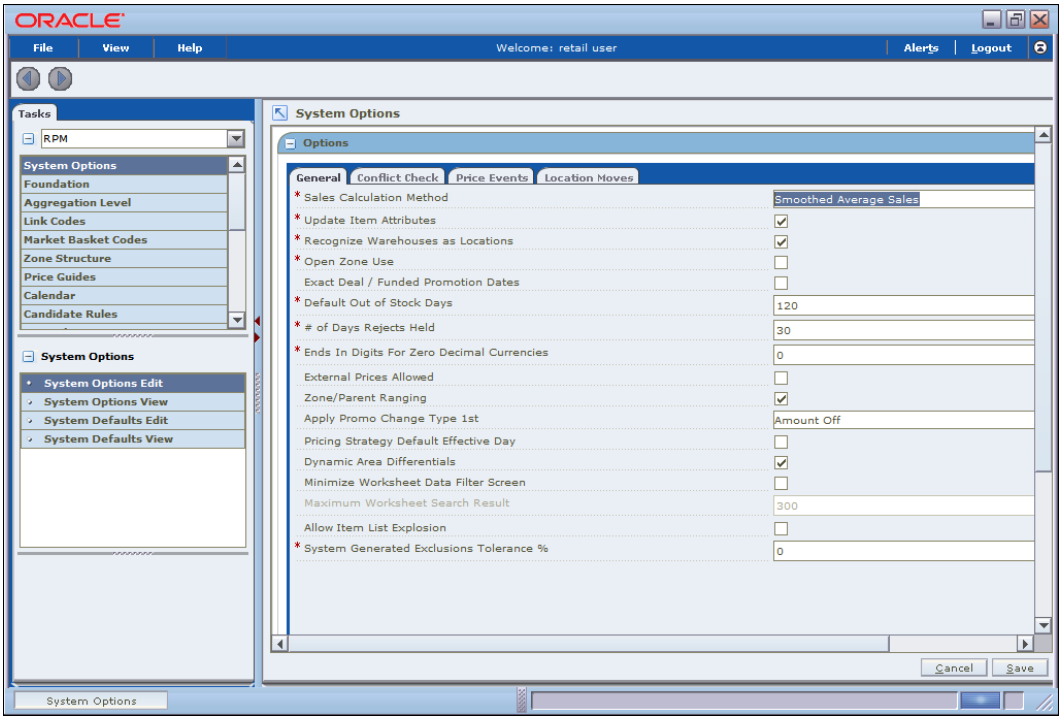
If problems occur when trying to start the RPM application, ensure proxies are turned off.

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

Example: http://appserver1:17011/rpm-client/launch?template=rpm_jnlp_template.vm

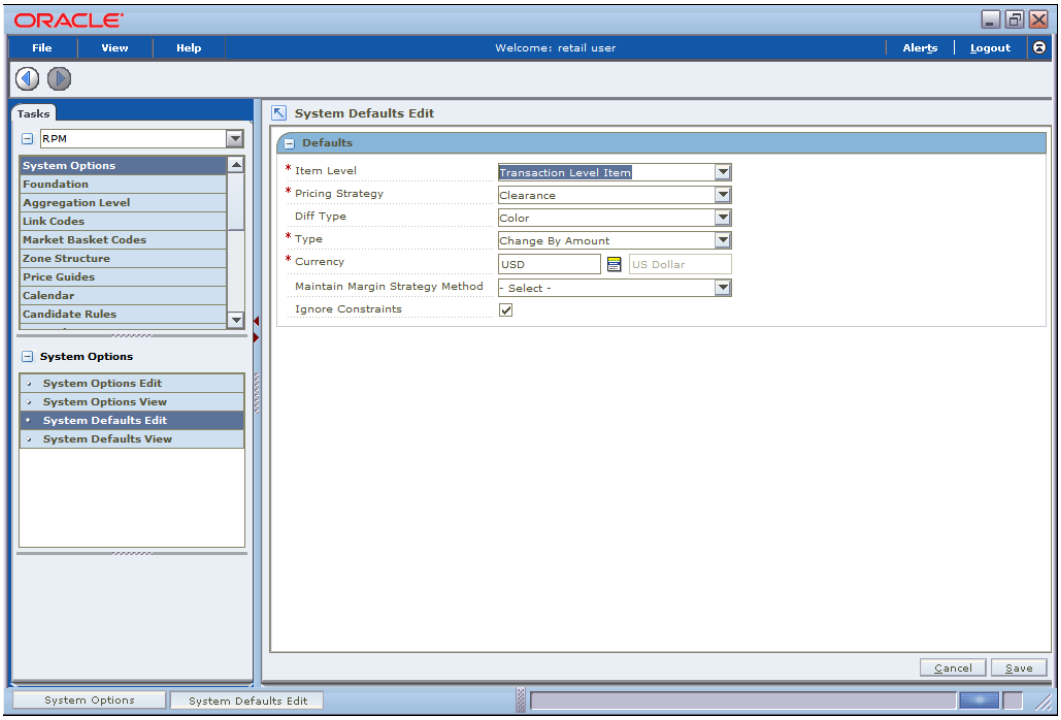
When you are in the RPM application, do the following to add a rpm_system_options row required by RPM for system use.

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



4. To add an rpm_system_**opt**ions_def row required by RPM needs for system use, do the following.

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



RPM Batch Scripts

The RPM application installer configures and installs the batch scripts under $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
priceEventItemListPurgeBatch.sh
priceEventPayloadPopulationBatch.sh
primaryZoneModificationsBatch.sh
promotionPriceChangePublishExport.sh
purgeExpiredExecutedOrApprovedClearancesBatch.sh
purgePayloadsBatch.sh
purgeUnusedAndAbandonedClearancesBatch.sh
regularPriceChangePublishExport.sh
RPMtoORPOSPublishBatch.sh
RPMtoORPOSPublishExport.sh
```

Example profile.sh

```
#!/bin/sh

#Need the Oracle Home set to aim at ORACLE Client or db on the server RPM
# is installed on
ORACLE_HOME=/u00/oracle/product/11.2.0.4

#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/rpm14/config/wallet
export TNS_ADMIN
    echo "ORACLE_HOME=${ORACLE_HOME}"
    echo "JAVA_HOME=${JAVA_HOME}"
    echo "PATH=${PATH}"
```

To source the profile above, do the following:

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

While running the 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.

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:

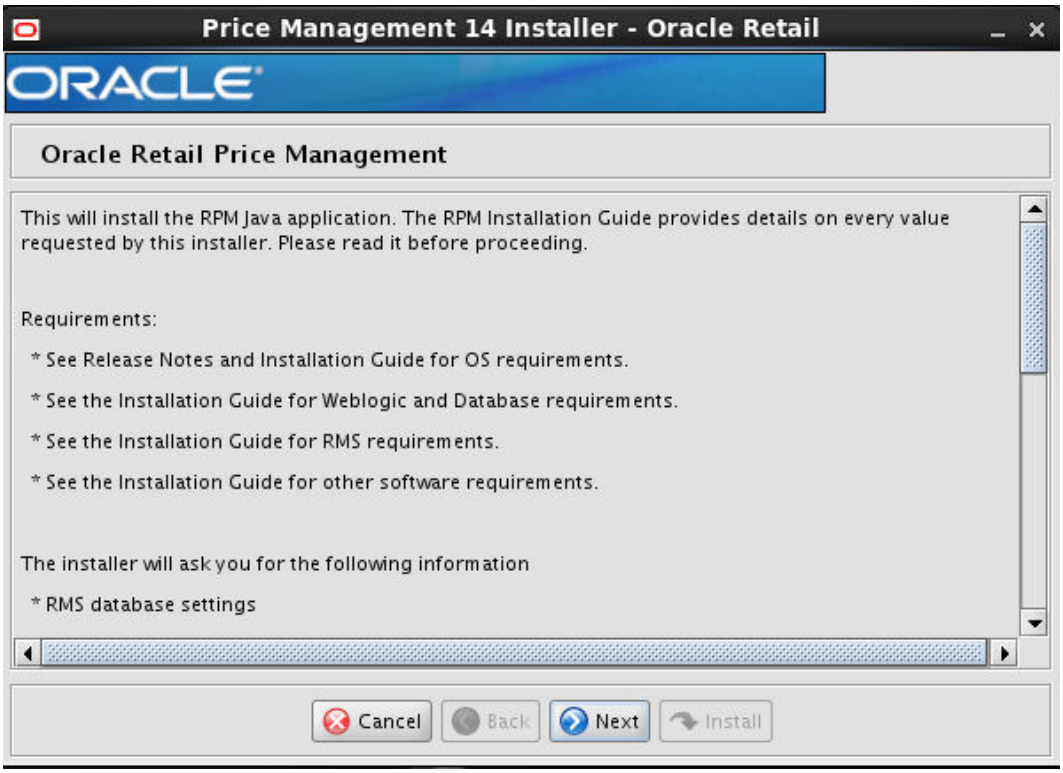
```
insert into rsm_user_role
(id, user_id, role_id, start_date_time, end_date_time)
select rsm_user_role_seq.nextval,
       'retail.user1',
       -1001,
       nvl(get_vdate,sysdate) - 365,
       null
  from dual;
```

A

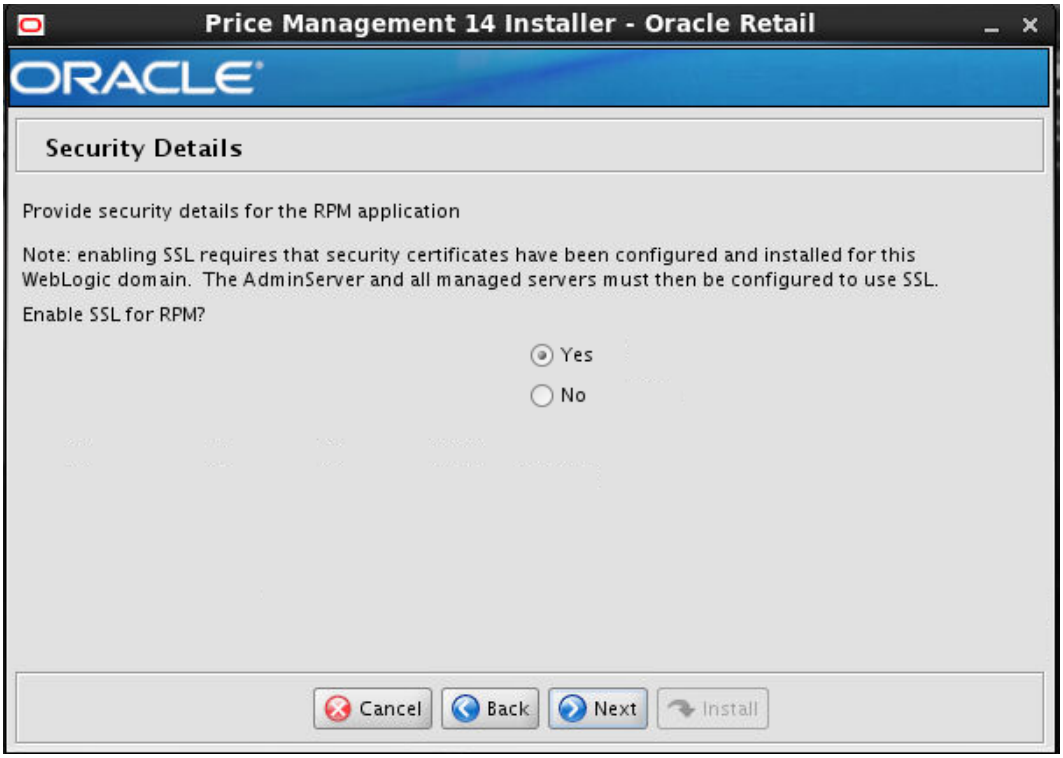
Appendix: RPM Application Installer Screens

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: Installation Introduction Screen

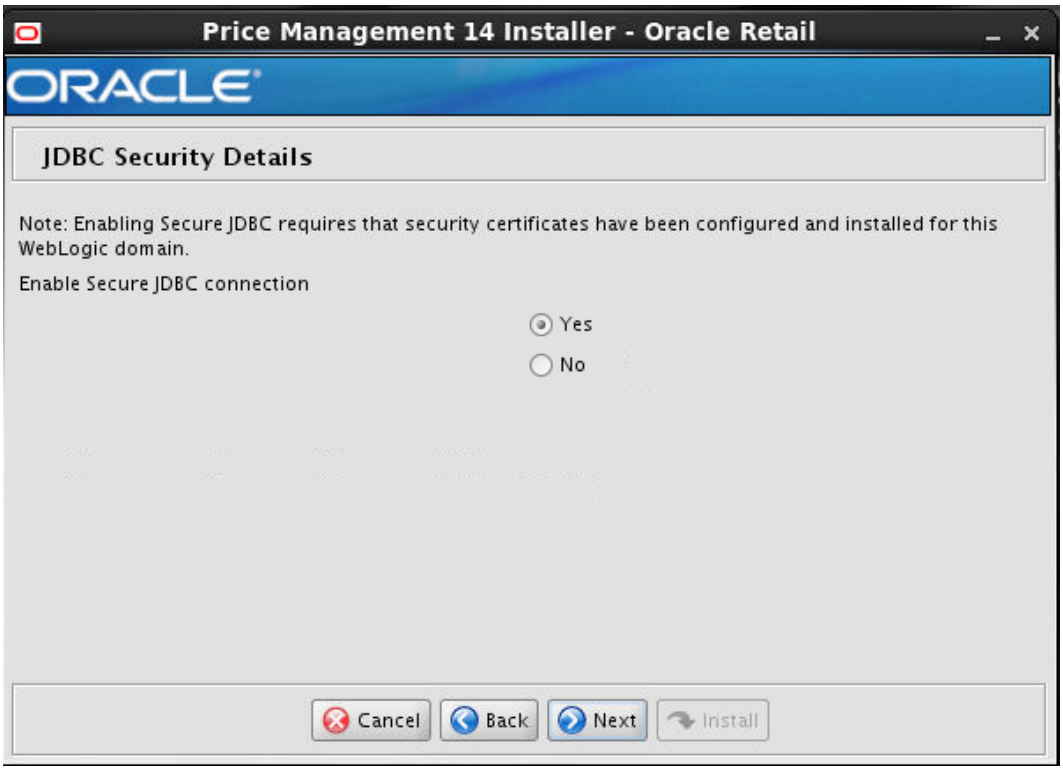


Screen: Security Details



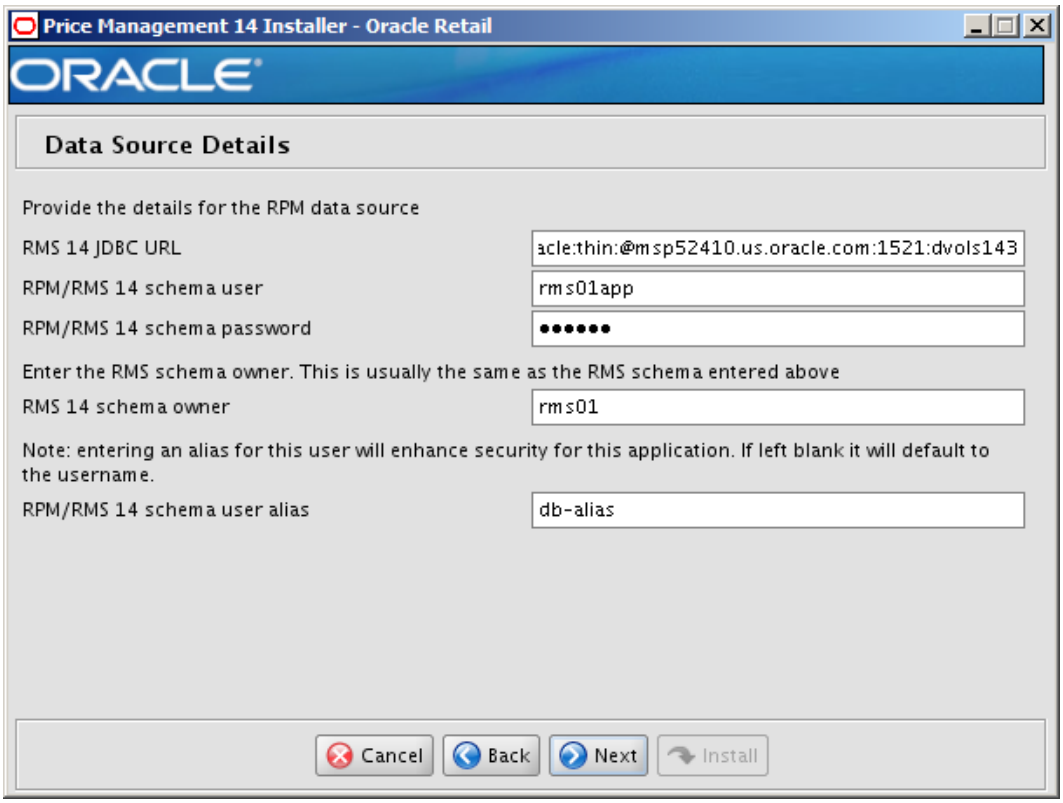
Field Title	Enable SSL for RPM?
Field Description	Choosing Yes will deploy RPM using SSL and configure RPM to use SSL. In this case, SSL must be configured and the ports must be enabled for the AdminServer and RPM managed servers. Choosing No will deploy and configure RPM without SSL. In this case the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

Screen: JDBC Security Details



Field Title	Enable Secure JDBC.?
Field Description	Choosing Yes will deploy the RPM data source with secure port and looks for truststore and keystore details of the RPM server and create the data source with the details

Screen: Data Source Details



Field Title	RMS 14 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	Weblogic server Administration Console.
Examples	jdbc:oracle:thin:@msp52410.us.oracle.com:1521:dvols143

Field Title	RPM/RMS 14 schema user
Field Description	Database user where the RMS database schema was installed.
Destination	Weblogic Administration Console – Data Source
Example	RMS01APP

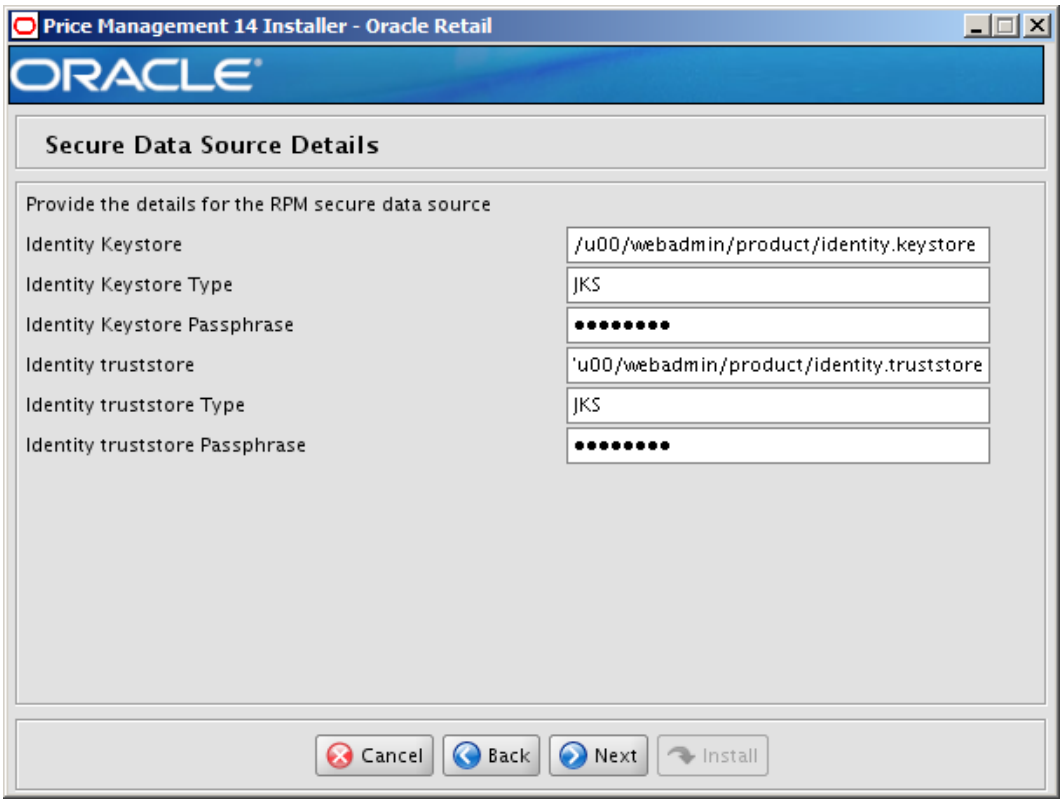
Field Title	RPM/RMS 14 schema password
Field Description	Password for the RMS schema user.
Destination	ORACLE java wallet file
Example	cwallet.sso

Field Title	RMS 14 schema owner
Field Description	Database user which owns the RMS tables. This is usually the same as the RMS 14 schema above.
Destination	rpm.properties
Example	RMS01

Field Title	RPM/RMS 14 schema alias
Field Description	Database user which owns the RMS tables. This is usually the same as the RMS 14 schema above.
Destination	rpm.properties and ORACLE java wallet file
Example	db-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: Secure Data Source Details

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



Field Title	Identity Keystore
Field Description	This is the path of the keystore which contains the ssl identity certificates of the host as obtained from the certificate authority.
Destination	Weblogic server Administration Console.
Example	/u00/webadmin/product/identity.keystore

Field Title	Identity Keystore Type
Field Description	The type of the keystore used
Destination	WebLogic server Administration Console.
Example	JKS

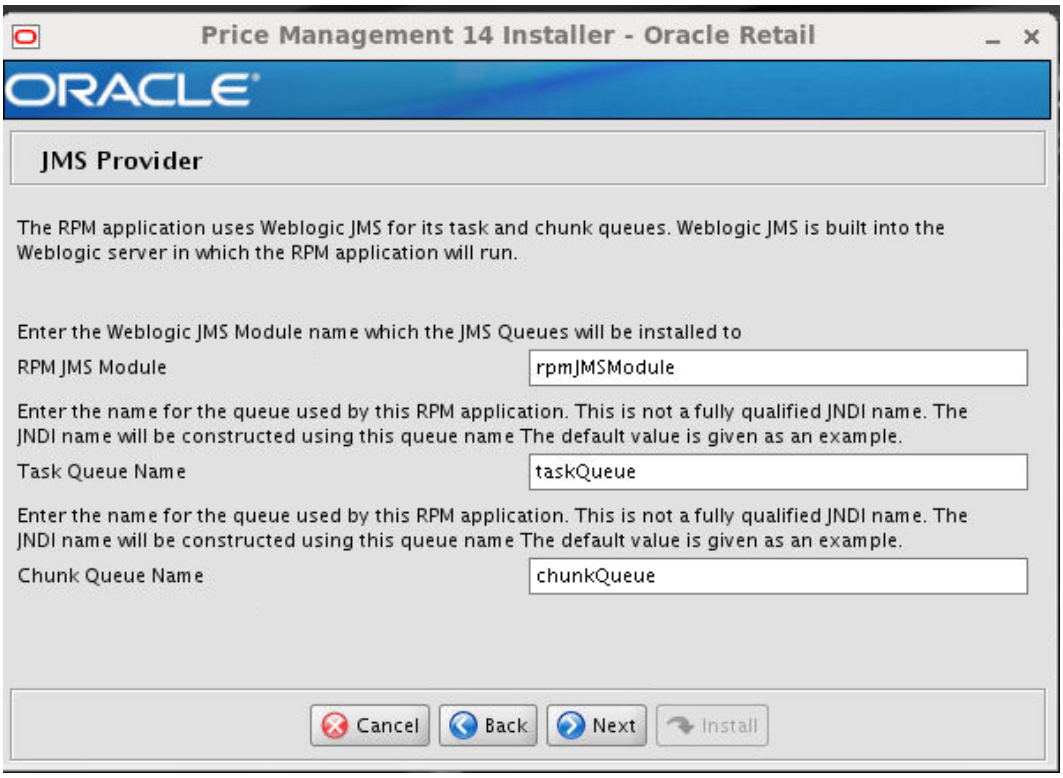
Field Title	Identity Keystore PassPhrase
Field Description	The password for the keystore used
Destination	WebLogic server Administration Console.

Field Title	Identity TrustStore
Field Description	This is the path of the keystore which contains the ssl root and optionally intermediate certificates as obtained from the certificate authority.
Destination	WebLogic server Administration Console.
Example	/u00/webadmin/product/identity.truststore

Field Title	Identity TrustStore Type
Field Description	The type of the truststore used
Destination	Weblogic server Administration Console.
Example	JKS

Field Title	Identity TrustStore PassPhrase
Field Description	The password of the truststore used
Destination	WebLogic server Administration Console.

Screen: JMS Provider

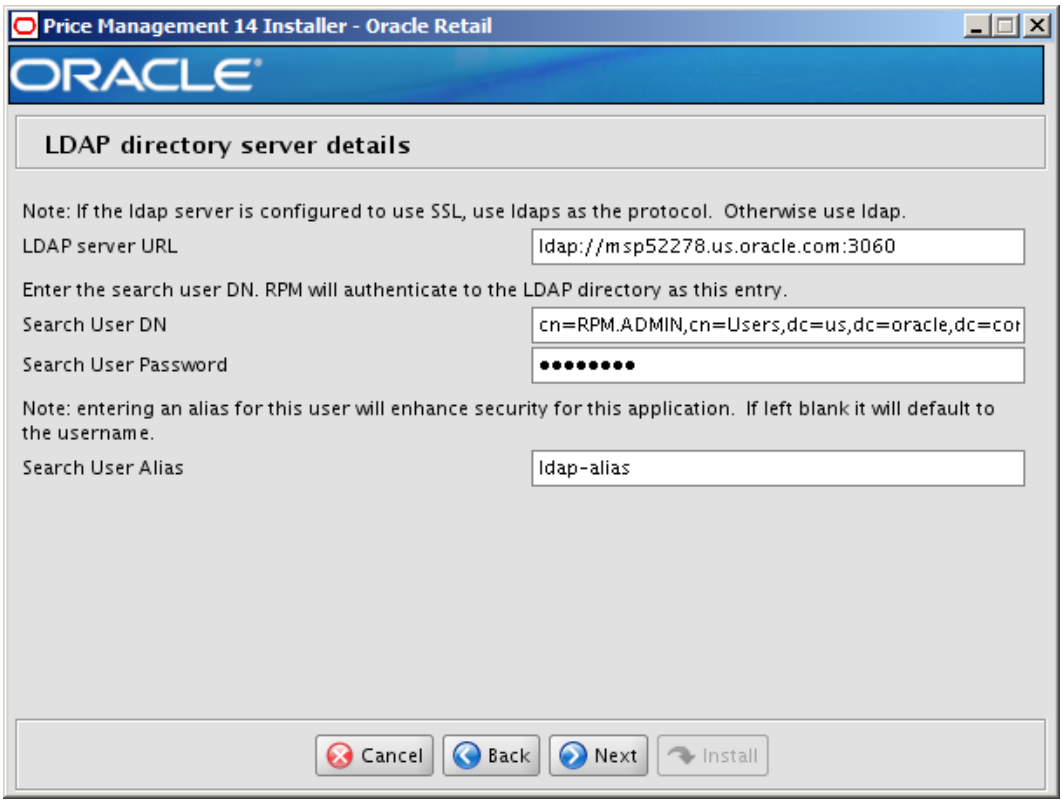


Field Title	RPM JMS Module
Field Description	The WebLogic JMS Module name to where the JMS Queues will be installed.
Destination	rpm.properties and WebLogic server Administration Console.
Example	rpmJMSModule

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).
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).
Destination	rpm.properties and WebLogic server Administration Console.
Example	chunkQueue

Screen: LDAP directory server details



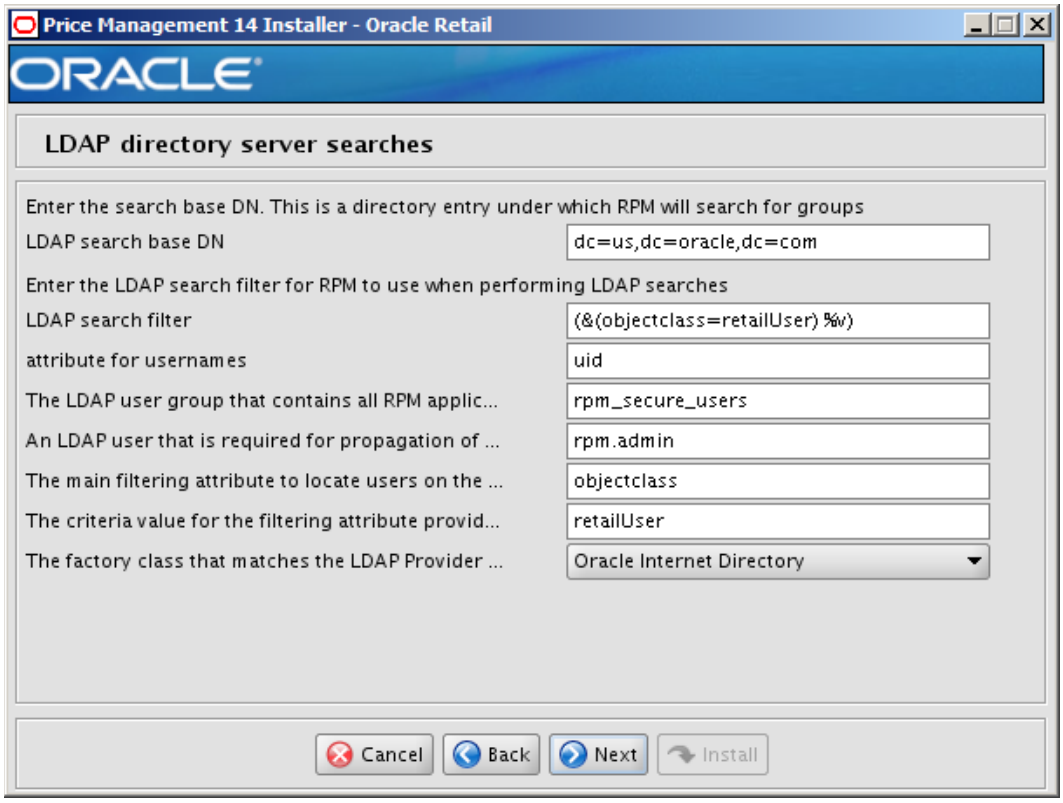
Field Title	LDAP server URL
Field Description	URL for your LDAP directory server. See Appendix: URL Reference for expected syntax.
Destination	security.properties
Example	ldap://myhost:3060/

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 (The details on creating this user is explained earlier in this document in the section 'Configure LDAP authentication Preinstallation Steps')

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

Field Title	The LDAP user group that contains all RPM application users.
Field Description	The LDAP user group that contains all RPM application users.
Destination	security.properties
Example	rpm_secure_users (This group is set up in the Groups DN Example: cn=Groups,dc=us,dc=oracle,dc=com. All Application users including cn=rpm.admin are added to the Members of this Group. Details of setting up this group is explained earlier in this guide in the section of 'Configure LDAP authentication Preinstallation Steps').

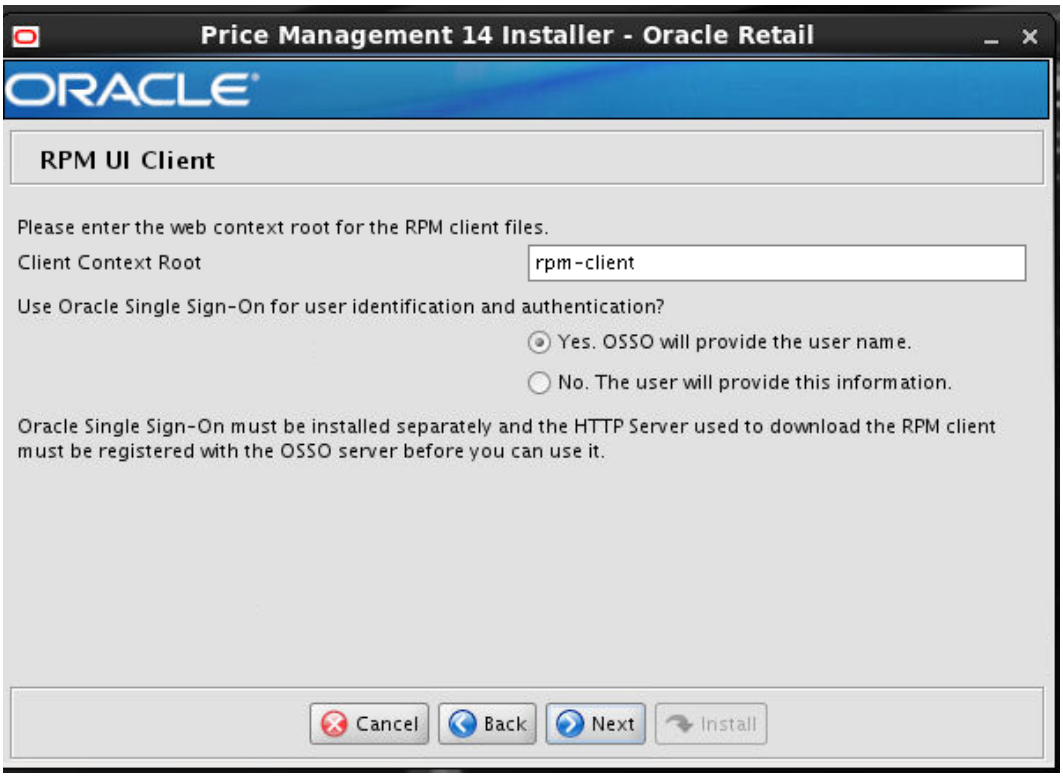
Field Title	An LDAP user that is required for propagation of security privileges. It must belong to the user group that contains all RPM application users.
Field Description	An LDAP user that is required for propagation of security privileges. It must belong to the user group that contains all RPM application users.
Destination	security.properties
Example	rpm.admin (The details on creating this user is explained earlier in this document in the section 'Configure LDAP authentication Preinstallation Steps')

Field Title	The main filtering attribute to locate users on the LDAP.
Field Description	The main filtering attribute to locate users on the LDAP.
Destination	security.properties
Example	objectclass

Field Title	The criteria value for the filtering attribute provided that valid users should match.
Field Description	The criteria value for the filtering attribute provided that valid users should match.
Destination	security.properties
Example	retailUser (This must be the same object class used as part of the 'LDAP search filter' field)

Field Title	The factory class that matches the LDAP Provider used to store identity information for the application.
Field Description	The factory class that matches the LDAP Provider used to store identity information for the application.
Destination	security.properties
Example	Oracle Internet Directory

Screen: RPM UI Client

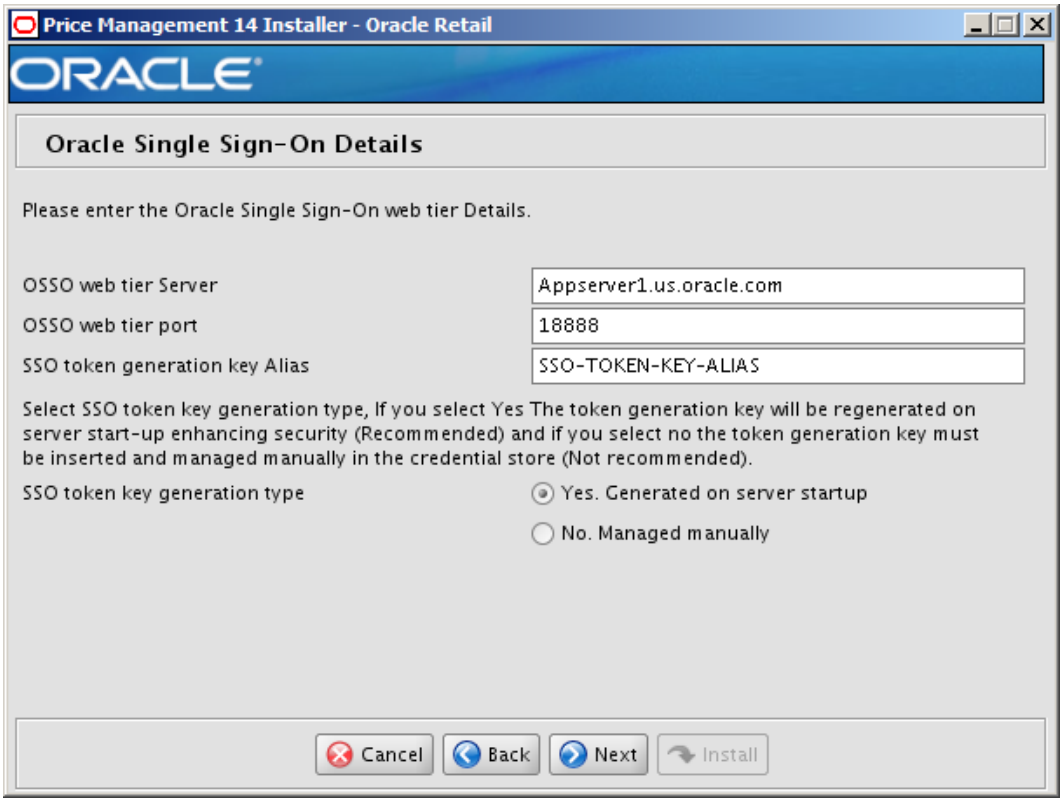


Field Title	Client Context Root
Field Description	The Client Context Root determines how the RPM client will be accessed from users' web browsers. The RPM client URL has the following format: http:/<host>:<port>/<rpm_client_ctx_root>/launch?template=rpm_jnlp_template.vm Example, with RPM Client Context Root value of rpm-client: http://appserver1:17011/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 Access Manager (Webgate Agent) technology to authenticate users. If OAM is being used in your environment, choose Yes. The No option configures RPM to use its own LDAP directory settings for authentication.
Example	No

Screen Single Sign-On Details

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



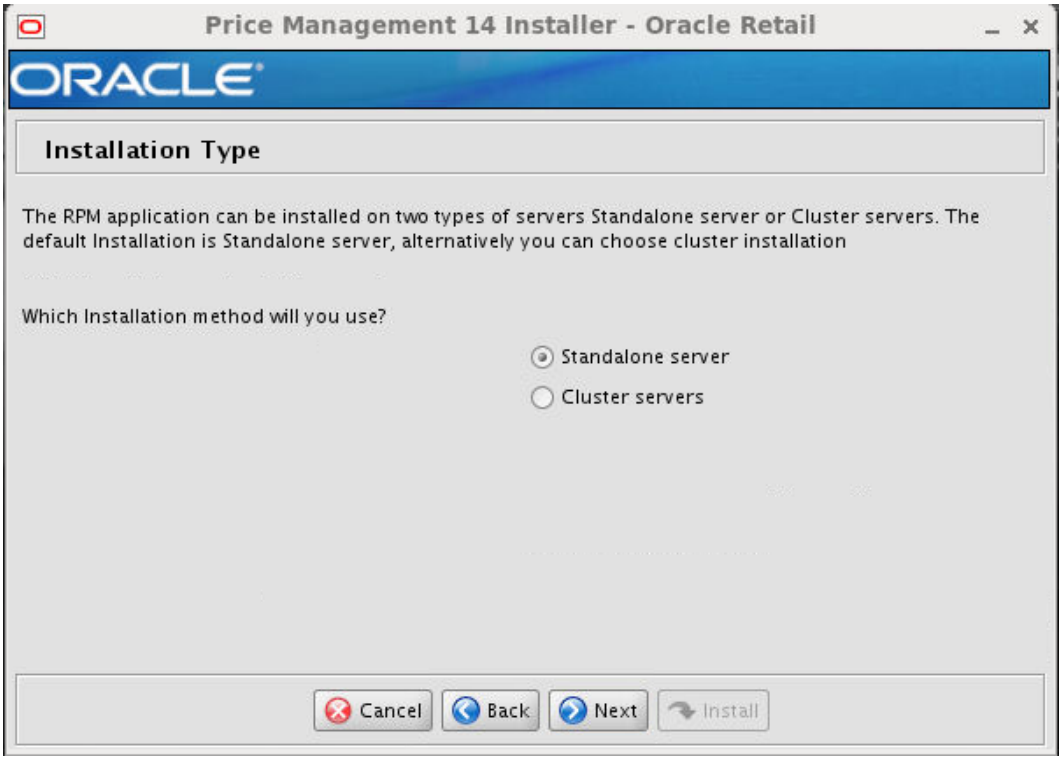
Field Title	OSSO Web Tier Server
Field Description	This should have the host name on which the web tier is deployed on
Example	Appserver1.us.oracle.com

Field Title	OSS Web Tier port
Field Description	The HTTP port of the webtier installation must be mentioned here.
Example	18888

Field Title	SSO Token generation key alias
Field Description	SSO uses this to store its tokens that is used to verify authenticity of the SSO call.
Example	SSO-TOKEN-KEY-ALIAS

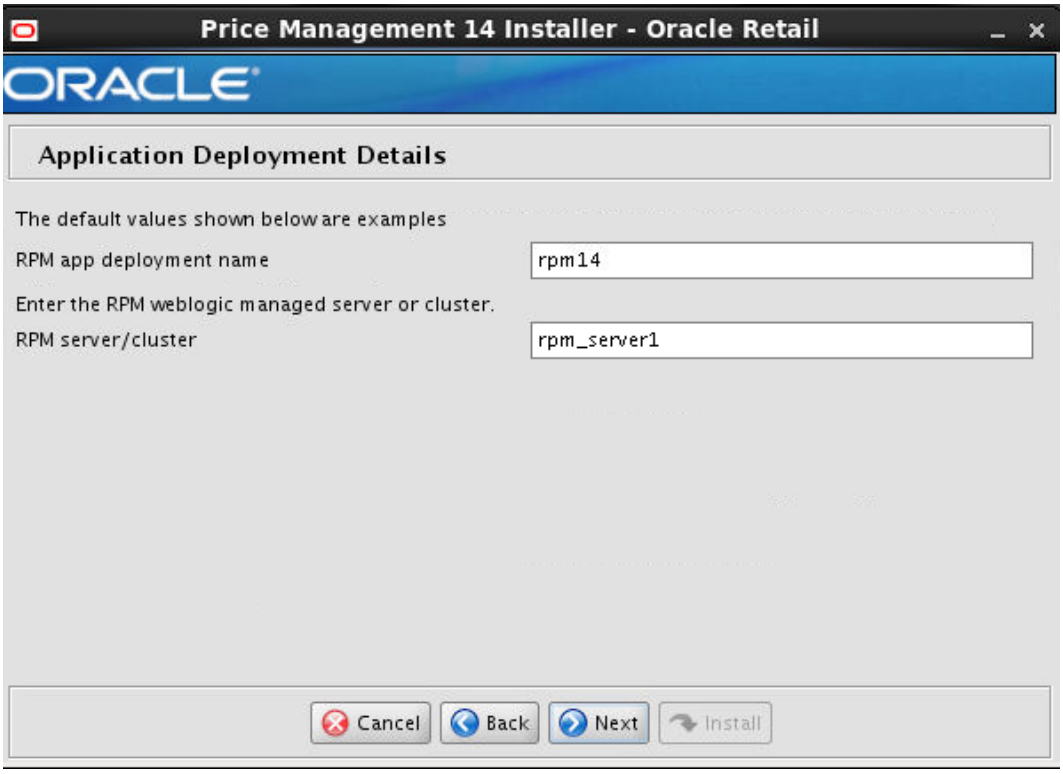
Field Title	SSO Token generation key type
Field Description	If you want to have the token generation by manual or by installer. If manual go for no. else select yes
Example	Yes

Screen Installation Type



Field Title	Installation type
Field Description	The default installation type is standalone server, alternatively you can choose cluster installation.

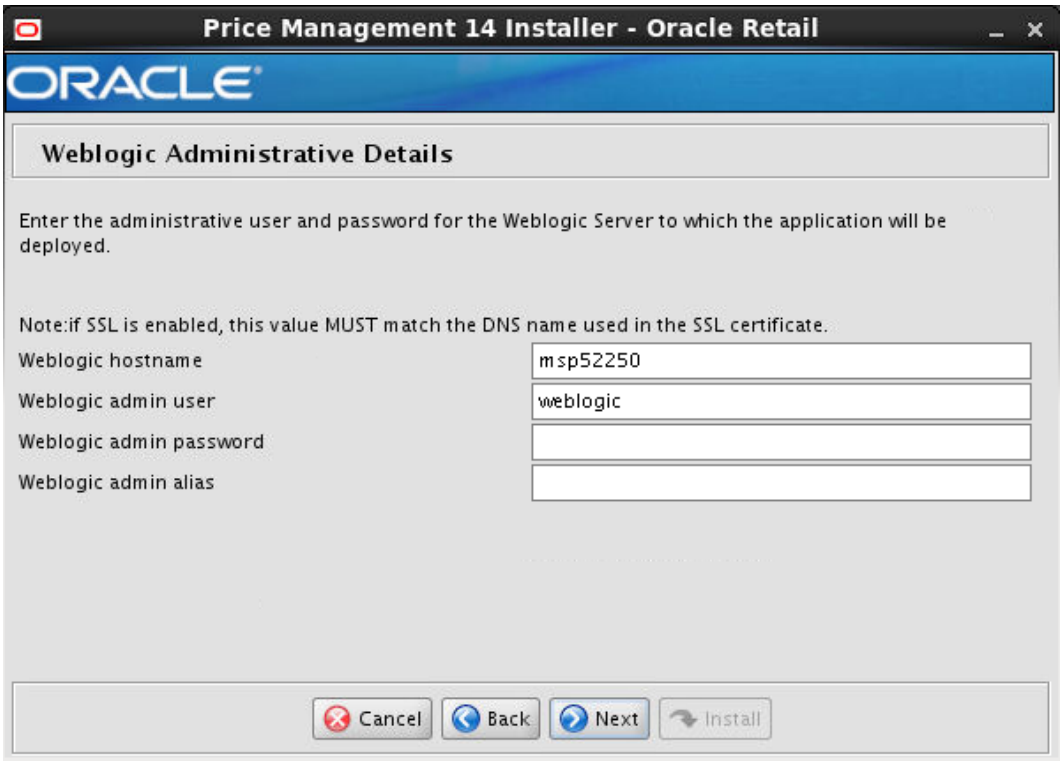
Screen: Application Deployment Details



Field Title	RPM 14 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	rpm14

Field Title	RPM 14 server/cluster
Field Description	Name of the server/cluster that was created for this RPM application. The deployment name given for the RPM 14 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



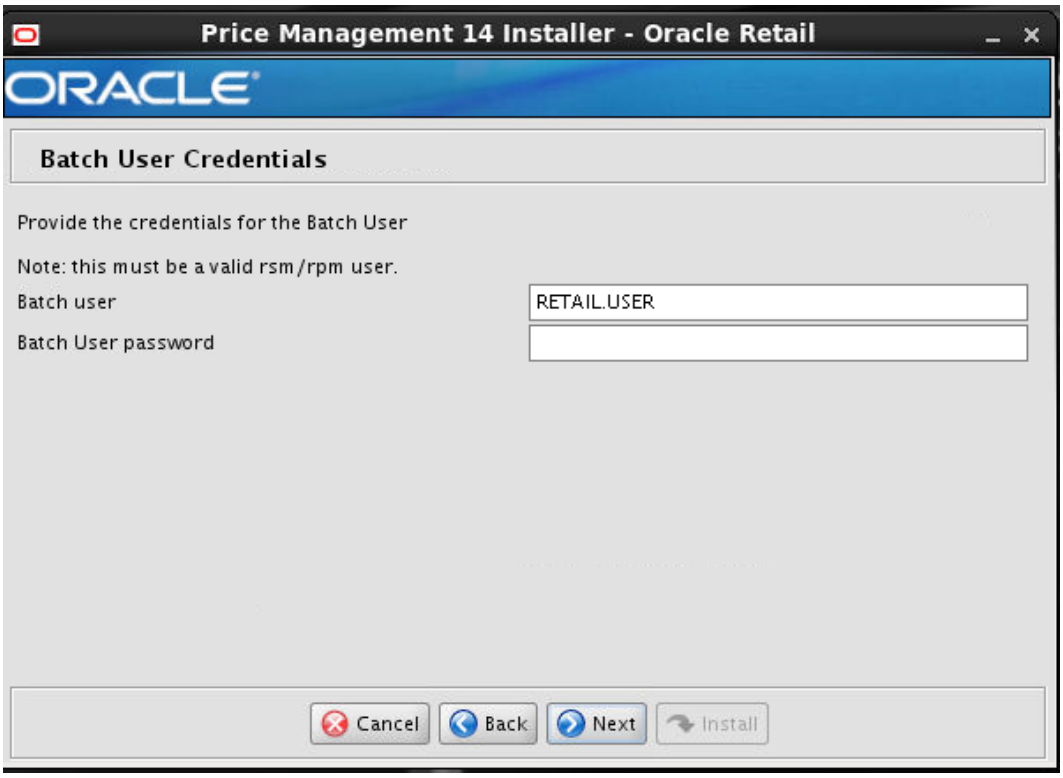
Field Title	Hostname
Field Description	Hostname of the application server. If SSL is used, this must match the DNS name in the SSL certificate.
Example	Myhost

Field Title	WebLogic admin user
Field Description	Username of the admin user for the WebLogic instance to which the RPM 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	weblogic-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 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

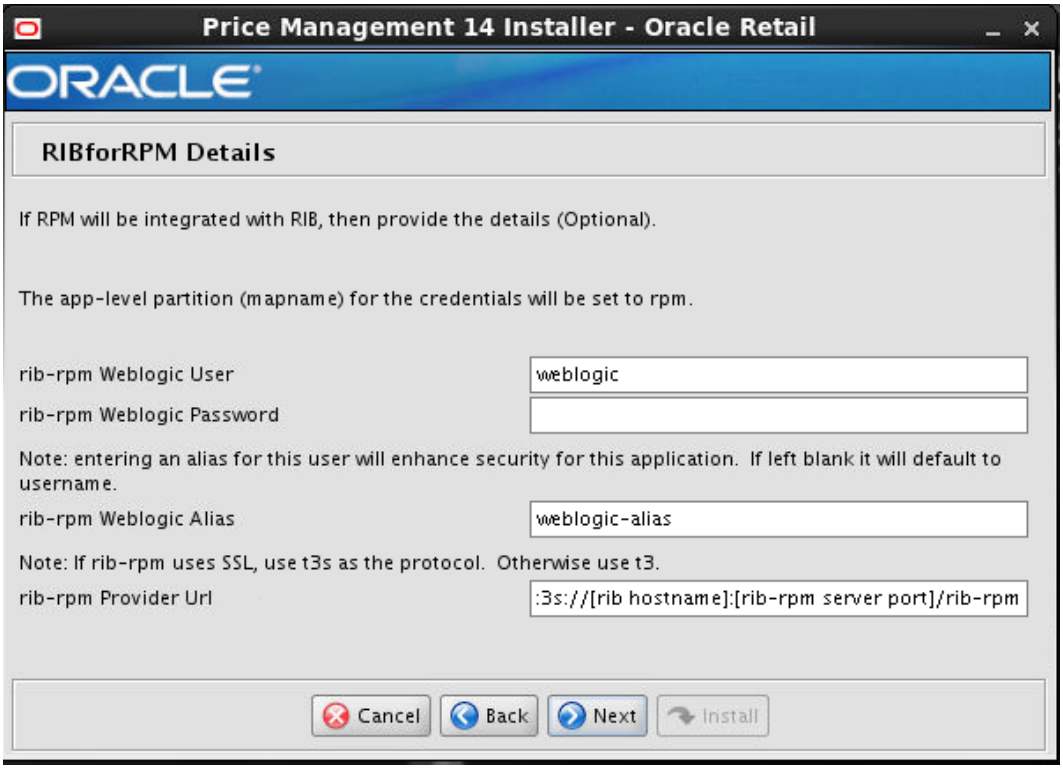
Field Title	Batch User Password
Field Description	The password of the batch user.

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

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



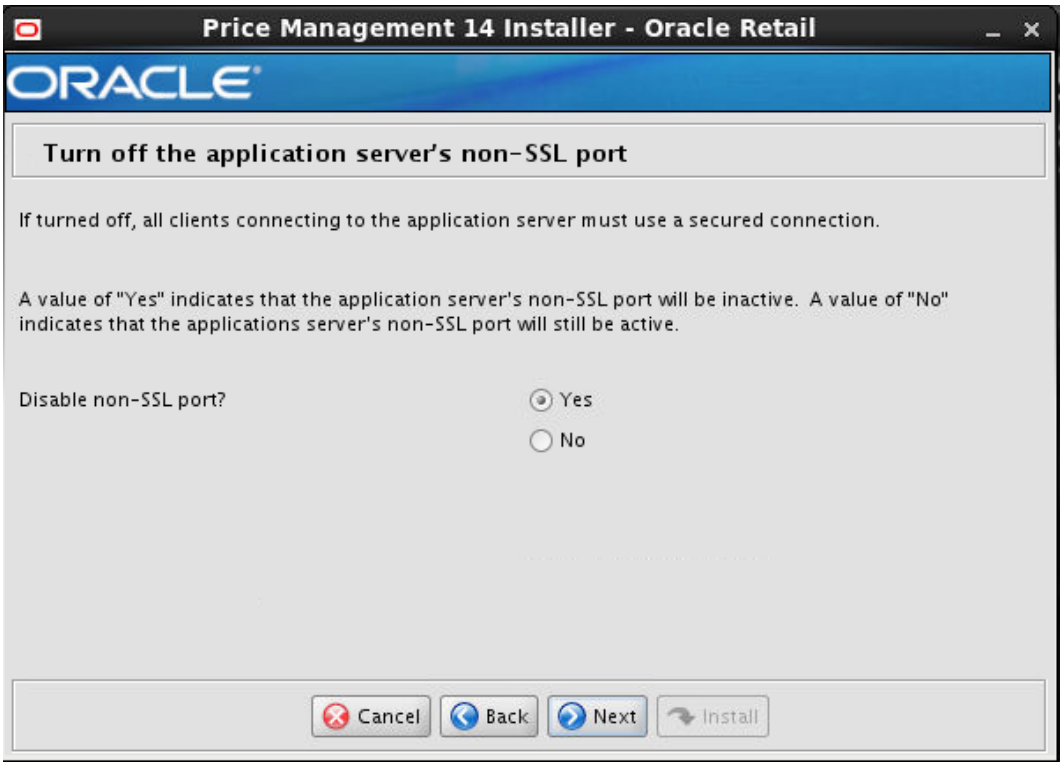
Field Title	rib-rpm WebLogic User
Field Description	The username of the rib-rpm WebLogic user.
Example	weblogic

Field Title	rib-rpm WebLogic password
Field Description	Password for the RIBforRPM 14 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.

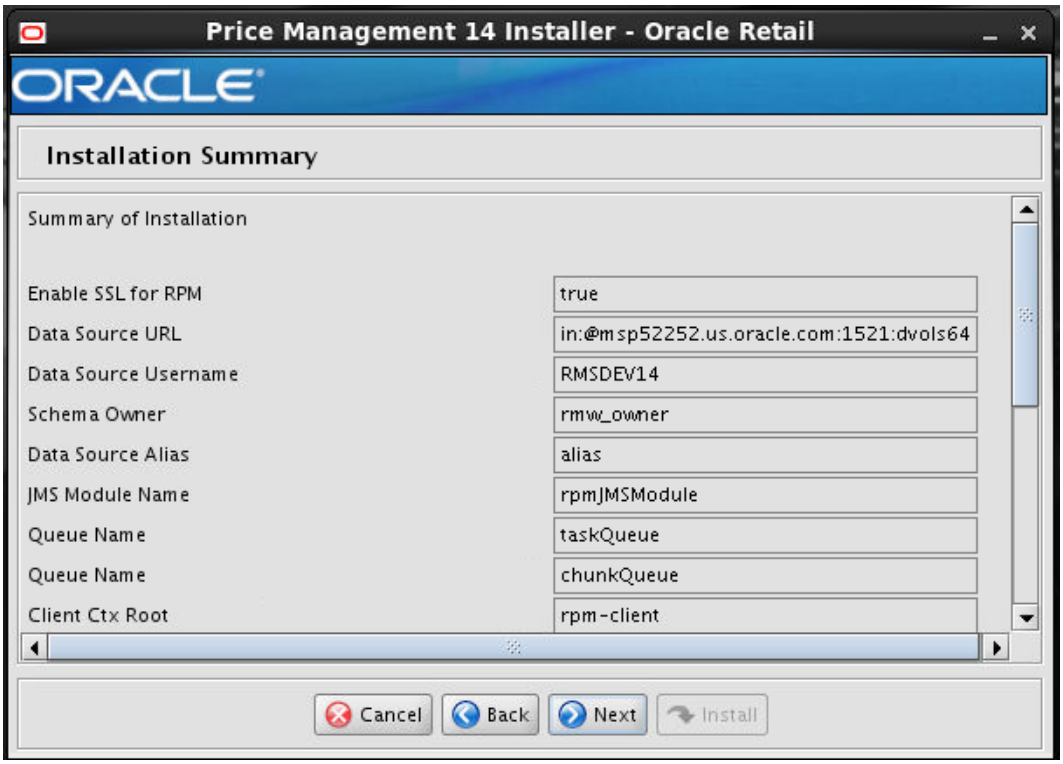
Field Title	rib-rpm Provider URL
Field Description	RPM provider URL for rib-rpm
Examples	t3s://myhost:8005/rib-rpm

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

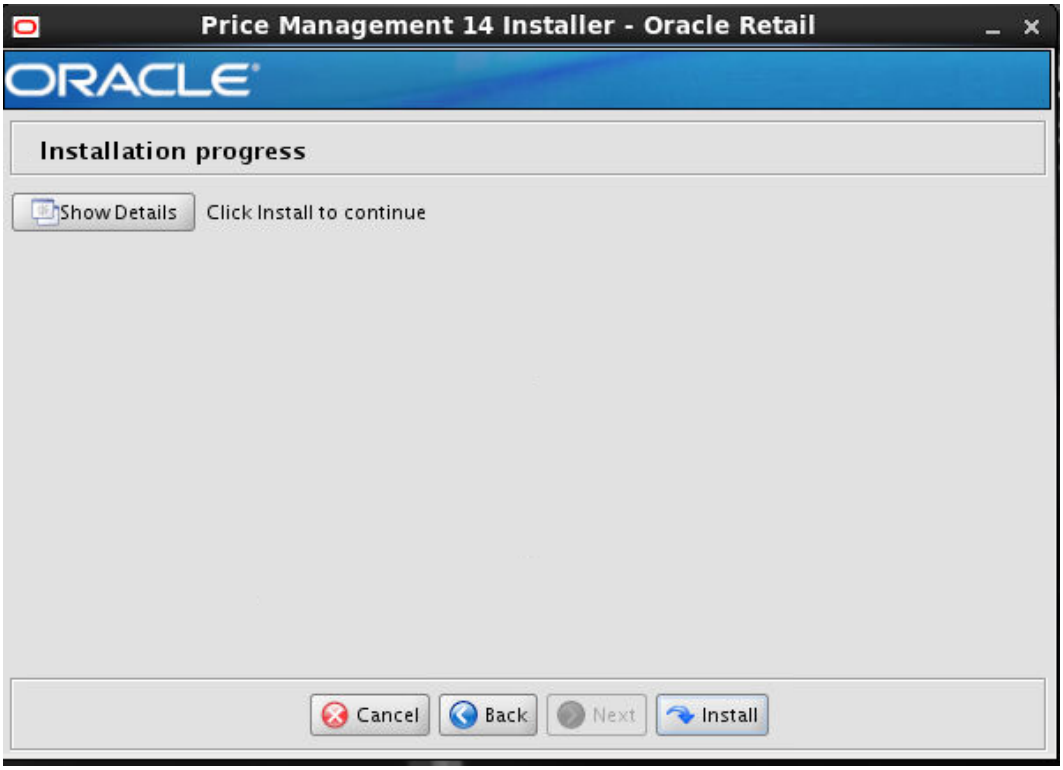


Field Title	Disable non-SSL port?
Field Description	Selecting the option 'Yes' indicates that the application server's non-SSL port will be inactive. Selecting "no" indicates that the application servers's non-SSL port will still be active This screen will appear only if you select SSL as Yes on the first screen

Screen: Installation Summary



Screen: Installation Progress



B

Appendix: Common Installation Errors

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

Keystore errors when signing rpm_client_config.jar

Error message

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

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

Solution

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

Unreadable buttons in the Installer

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

Left menu buttons missing in RPM Client

Symptom

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

Solution

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

Warning: Could not find X Input Context

Symptom

The following text appears in the console window during execution of the installer in GUI mode:

```
Couldn't find X Input Context
```

Solution

This message is harmless and can be ignored.

Failed RPM Login

Symptom

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

Solution

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

```
-Djava.security.auth.login.config=<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.

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.

C

Appendix: URL Reference

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

JDBC URL for a Database

Used by the Java application and by the installer to connect to the database.

Thick Client Syntax: jdbc:oracle:oci:@<sid>

<sid>: system identifier for the database

> **Example:** jdbc:oracle:oci:@mysid

Thin Client Syntax: jdbc:oracle:thin:@<host>:<port>:<sid>

<host>: hostname of the database server

<port>: database listener port

<sid>: system identifier for the database

> **Example:** jdbc:oracle:thin:@myhost:1521:mysid

JNDI Provider URL for an Application

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

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

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

> **Example:** t3:/myhost:17011/rpm14
>
> **Note:** The JNDI provider URL can have a different format depending on your cluster topology. Consult the WebLogic documentation.

D

Appendix: Setting Up Password Stores with wallets/credential stores

As part of an application installation, administrators must set up password stores for user accounts using wallets/credential stores. Some password stores must be installed on the application database side. While the installer handles much of this process, the administrators must perform some additional steps.

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

Oracle Retail Merchandising applications now have three different types of password stores. They are database wallets, java wallets, and database credential stores. Background and how to administer them below are explained in this appendix

About Database Password Stores and Oracle Wallet

Oracle databases have allowed other users on the server to see passwords in case database connect strings (username/password@db) were passed to programs. In the past, users could navigate to `ps -ef|grep <username>` to see the password if the password was supplied in the command line when calling a program.

To make passwords more secure, Oracle Retail has implemented the Oracle Software Security Assurance (OSSA) program. Sensitive information such as user credentials now must be encrypted and stored in a secure location. This location is called password stores or wallets. These password stores are secure software containers that store the encrypted user credentials.

Users can retrieve the credentials using aliases that were set up when encrypting and storing the user credentials in the password store. For example, if `username/password@db` is entered in the command line argument and the alias is called `db_username`, the argument to a program is as follows:

```
sqlplus /@db_username
```

This would connect to the database as it did previously, but it would hide the password from any system user.

After this is configured, as in the example above, the application installation and the other relevant scripts are no longer needed to use embedded usernames and passwords. This reduces any security risks that may exist because usernames and passwords are no longer exposed.

When the installation starts, all the necessary user credentials are retrieved from the Oracle Wallet based on the alias name associated with the user credentials.

There are three different types of password stores. One type explain in the next section is for database connect strings used in program arguments (such as `sqlplus /@db_username`). The others are for Java application installation and application use.

Setting Up Password Stores for Database User Accounts

After the database is installed and the default database user accounts are set up, administrators must set up a password store using the Oracle wallet. This involves

assigning an alias for the username and associated password for each database user account. The alias is used later during the application installation. This password store must be created on the system where the application server and database client are installed.

This section describes the steps you must take to set up a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, see the *Oracle Database Security Guide*.

> **Note:** In this section, <wallet_location> is a placeholder text for illustration purposes. Before running the command, ensure that you specify the path to the location where you want to create and store the wallet.

To set up a password store for the database user accounts, perform the following steps:

1. Create a wallet using the following command:

   ```
   mkstore -wrl <wallet_location> -create
   ```

   After you run the command, a prompt appears. Enter a password for the Oracle Wallet in the prompt.

   > **Note:** The mkstore utility is included in the Oracle Database Client installation.

   The wallet is created with the auto-login feature enabled. This feature enables the database client to access the wallet contents without using the password. For more information, refer to the *Oracle Database Advanced Security Administrator's Guide.*

2. Create the database connection credentials in the wallet using the following command:

   ```
   mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>
   ```

   After you run the command, a prompt appears. Enter the password associated with the database user account in the prompt.

3. Repeat Step 2 for all the database user accounts.

4. Update the sqlnet.ora file to include the following statements:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = <wallet_location>)))
   SQLNET.WALLET_OVERRIDE = TRUE
   SSL_CLIENT_AUTHENTICATION = FALSE
   ```

5. Update the tnsnames.ora file to include the following entry for each alias name to be set up.

   ```
   <alias-name> =
       (DESCRIPTION =
        (ADDRESS_LIST =
               (ADDRESS = (PROTOCOL = TCP) (HOST = <host>) (PORT = <port>))
           )
           (CONNECT_DATA =
               (SERVICE_NAME = <service>)
            )
        )
   ```

   In the previous example, <alias-name>, <host>, <port>, and <service> are placeholder text for illustration purposes. Ensure that you replace these with the relevant values.

Setting up Wallets for Database User Accounts

The following examples show how to set up wallets for database user accounts for the following applications:

- For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI

For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI

To set up wallets for database user accounts, do the following.

1. Create a new directory called wallet under your folder structure.

```
cd /projects/rms14/dev/
mkdir .wallet
```

> **Note:** The default permissions of the wallet allow only the owner to use it, ensuring the connection information is protected. If you want other users to be able to use the connection, you must adjust permissions appropriately to ensure only authorized users have access to the wallet.

2. Create a sqlnet.ora in the wallet directory with the following content.

```
WALLET_LOCATION =   (SOURCE =     (METHOD = FILE)     (METHOD_DATA =
(DIRECTORY =  /projects/rms14/dev/.wallet)) )
SQLNET.WALLET_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```

> **Note**: WALLET_LOCATION must be on line 1 in the file.

3. Setup a tnsnames.ora in the wallet directory. This tnsnames.ora includes the standard tnsnames.ora file. Then, add two custom tns_alias entries that are only for use with the wallet. For example, `sqlplus /@dvols29_rms01user`.

```
ifile = /u00/oracle/product/11.2.0.1/network/admin/tnsnames.ora

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

dvols29_rms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspxxxxx.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/rms14/dev/.wallet
```

   b. Create the wallet files.

```
$ mkstore -wrl . –create
```

   c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.

d. Enter the password again.

Two wallet files are created from the above command:

- ewallet.p12
- cwallet.sso

5. Create the wallet entry that associates the user name and password to the custom tns alias that was setup in the wallet's tnsnames.ora file.

```
mkstore –wrl . –createCredential <tns_alias> <username> <password>
```

Example: `mkstore –wrl . –createCredential dvols29_rms01user rms01user passwd`

6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.

```
$ export TNS_ADMIN=/projects/rms14/dev/.wallet /* This is very import to use
wallet to point at the alternate tnsnames.ora created in this example */

$ sqlplus /@dvols29_rms01user

SQL*Plus: Release 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 the value of the entry:

  ```
  rms01user
  ```

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

  Returns the value of the entry:

  ```
  Passwd
  ```

Setting up RETL Wallets

RETL creates a wallet under $RFX_HOME/etc/security, with the following files:

- cwallet.sso
- jazn-data.xml
- jps-config.xml
- README.txt

To set up RETL wallets, perform the following steps:

1. Set the following environment variables:
   - `ORACLE_SID=<retaildb>`
   - `RFX_HOME=/u00/rfx/rfx-13`
   - `RFX_TMP=/u00/rfx/rfx-13/tmp`
   - `JAVA_HOME=/usr/jdk1.6.0_12.64bit`
   - `LD_LIBRARY_PATH=$ORACLE_HOME`
   - `PATH=$RFX_HOME/bin:$JAVA_HOME/bin:$PATH`
2. Change directory to $RFX_HOME/bin.
3. Run setup-security-credential.sh.
   - Enter 1 to add a new database credential.
   - Enter the dbuseralias. For example, `retl_java_rms01user`.
   - Enter the database user name. For example, `rms01user.`
   - Enter the database password.
   - Re-enter the database password.
   - Enter D to exit the setup script.
4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

   For example, to configure RETLforRPAS, modify the following entries in `$MMHOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env.`
   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     - `export RETL_WALLET_ALIAS="retl_java_rms01user"`
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     - `export ORACLE_WALLET_ALIAS="dvols29_rms01user"`
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     - `export SQLPLUS_LOGON="/@${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.

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

For Java applications, consider the following:

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.
- Passwords are not used to update entries in Java wallets. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application that has been installed, the wallet is located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/config Example: /u00/webadmin/product/10.3.6/WLS/user_projects/domains/14_mck_soa_domain/retail/reim14/config
- Application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.
- Scripts are located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/retail-public-security-api/bin for administering wallet entries.
- Example:
- /u00/webadmin/product/10.3.6/WLS/user_projects/domains/REIMDomain/retail/reim14/retail-public-security-api/bin
- In this directory is a script to help you update each alias entry without having to remember the wallet details. For example, if you set the RPM database alias to rms01user, you will find a script called update-RMS01USER.sh.

  Note: These scripts are available only with applications installed by way of an installer.

- Two main scripts are related to this script in the folder for more generic wallet operations: dump_credentials.sh and save_credential.sh.
- If you have not installed the application yet, you can unzip the application zip file and view these scripts in <app>/application/retail-public-security-api/bin.
- Example:
- /u00/webadmin/reim14/application/retail-public-security-api/bin

update-<ALIAS>.sh

update-<ALIAS>.sh updates the wallet entry for this alias. You can use this script to change the user name and password for this alias. Because the application refers only to the alias, no changes are needed in application properties files.

Usage:

```
update-<username>.sh <myuser>
```

Example:

```
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/reta
il-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/RPMDomain/retail/rpm14/reta
il-public-security-api/bin>
```

dump_credentials.sh

dump_credentials.sh is used to retrieve information from wallet. For each entry found in the wallet, the wallet partition, the alias, and the user name are displayed. Note that the password is not displayed. If the value of an entry is uncertain, run save_credential.sh to resave the entry with a known password.

```
dump_credentials.sh <wallet location>
```

Example:

```
dump_credentials.sh
location:/u00/webadmin/product/10.3.x/WLS/user_projects/domains/REIMDomain/retail/
reim14/config

Retail Public Security API Utility
==========================================
```

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

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

Application level key partition name:reim14
User Name Alias:WLS-ALIAS User Name:weblogic
User Name Alias:RETAIL-ALIAS User Name:retail.user
User Name Alias:LDAP-ALIAS User Name:RETAIL.USER
User Name Alias:RMS-ALIAS User Name:rms14mock
User Name Alias:REIMBAT-ALIAS User Name:reimbat
```

save_credential.sh

save_credential.sh is used to update the information in wallet. If you are unsure about the information that is currently in the wallet, use dump_credentials.sh as indicated above.

```
save_credential.sh -a <alias> -u <user> -p <partition name>  –l <path of the
wallet file location where credentials are stored>
```

Example:

```
mspdv351:[1036_WLS] /u00/webadmin/mock14_testing/rtil/rtil/application/retail-
public-security-api/bin> save_credential.sh -l wallet_test -a myalias -p
mypartition -u myuser

==========================================
Retail Public Security API Utility
==========================================

Enter password:
Verify password:
```

> **Note:** -p in the above command is for partition name. You must specify the proper partition name used in application code for each Java application.
>
> save_credential.sh and dump_credentials.sh scripts are the same for all applications. If using save_credential.sh to add a wallet entry or to update a wallet entry, bounce the application/managed server so that your changes are visible to the application. Also, save a backup copy of your cwallet.sso file in a location outside of the deployment path, because redeployment or reinstallation of the application will wipe the wallet entries you made after installation of the application. To restore your wallet entries after a redeployment/reinstallation, copy the backed up cwallet.sso file over the cwallet.sso file. Then bounce the application/managed server.

Usage

```
==========================================
Retail Public Security API Utility
==========================================
usage: save_credential.sh -au[plh]
E.g. save_credential.sh -a rms-alias -u rms_user -p rib-rms -l ./
 -a,--userNameAlias <arg>              alias for which the credentials
needs to be stored
 -h,--help                             usage information
 -l,--locationofWalletDir <arg>        location where the wallet file is
created.If not specified, it creates the wallet under secure-credential-wallet
directory which is already present under the retail-public-security-api/
directory.
 -p,--appLevelKeyPartitionName <arg>   application level key partition name
 -u,--userName <arg>                   username to be stored in secure
credential wallet for specified alias*
```

How does the Wallet Relate to the Application?

The ORACLE Retail Java applications have the wallet alias information you create in an <app-name>.properties file. Below is the reim.properties file. Note the database information and the user are presented as well. The property called datasource.credential.alias=RMS-ALIAS uses the ORACLE wallet with the argument of RMS-ALIAS at the csm.wallet.path and csm.wallet.partition.name = reim14 to retrieve the password for application use.

Reim.properties code sample:

```
datasource.url=jdbc:oracle:thin:@mspxxxxx.us.oracle.com:1521:pkols07
datasource.schema.owner=rms14mock
datasource.credential.alias=RMS-ALIAS
# ===============================================================
# ossa related Configuration
#
# These settings are for ossa configuration to store credentials.
# ===============================================================

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

How does the Wallet Relate to Java Batch Program use?

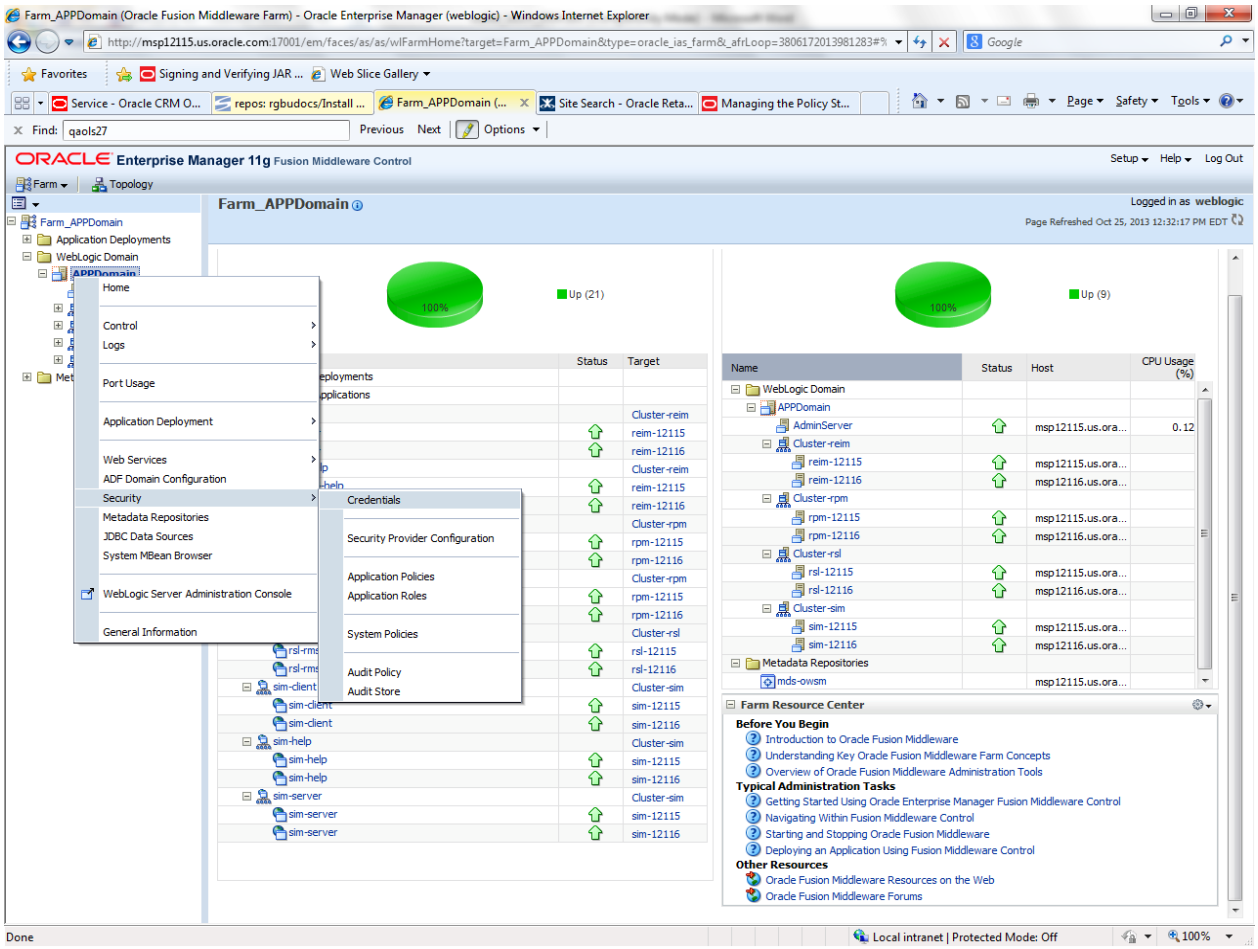
Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser RMS01APP, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments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

Database Credential Store Administration

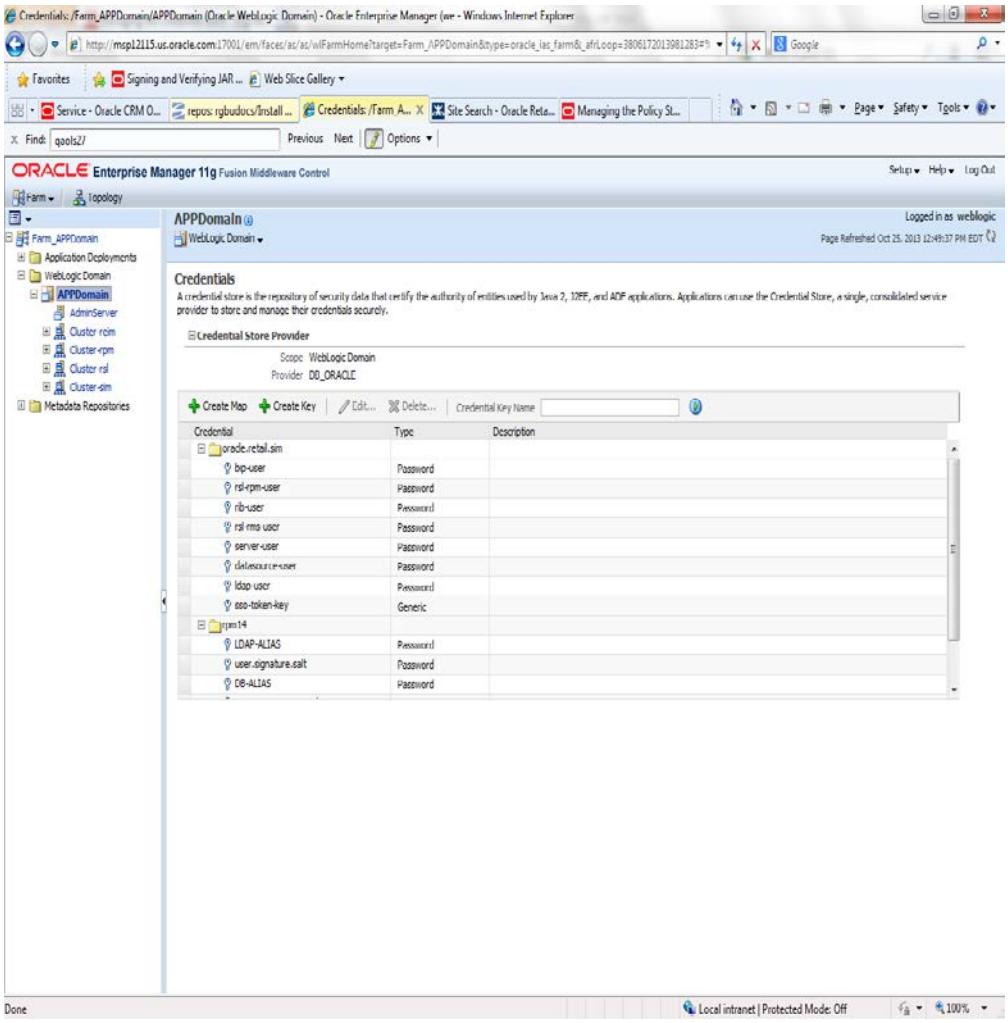
ORACLE Retail 14.0.2 brings something new into the password stores. A domain level database credential store. This is used in RPM login processing, SIM login processing, and Allocation login processing and policy information for application permission. Setting up the database credential store is addressed in the RPM, SIM, and Alloc 14.0.2 install guides.

The following sections show an example of how to administer the password stores thru ORACLE Enterprise Manger Fusion Middleware Control, a later section will show how to do this thru WLST scripts.

1. The first step is to use your link to Oracle Enterprise Manager Fusion Middleware Control for the domain in question. Locate your domain on the left side of the screen and do a right mouse click on the domain and select **Security** > **Credentials**

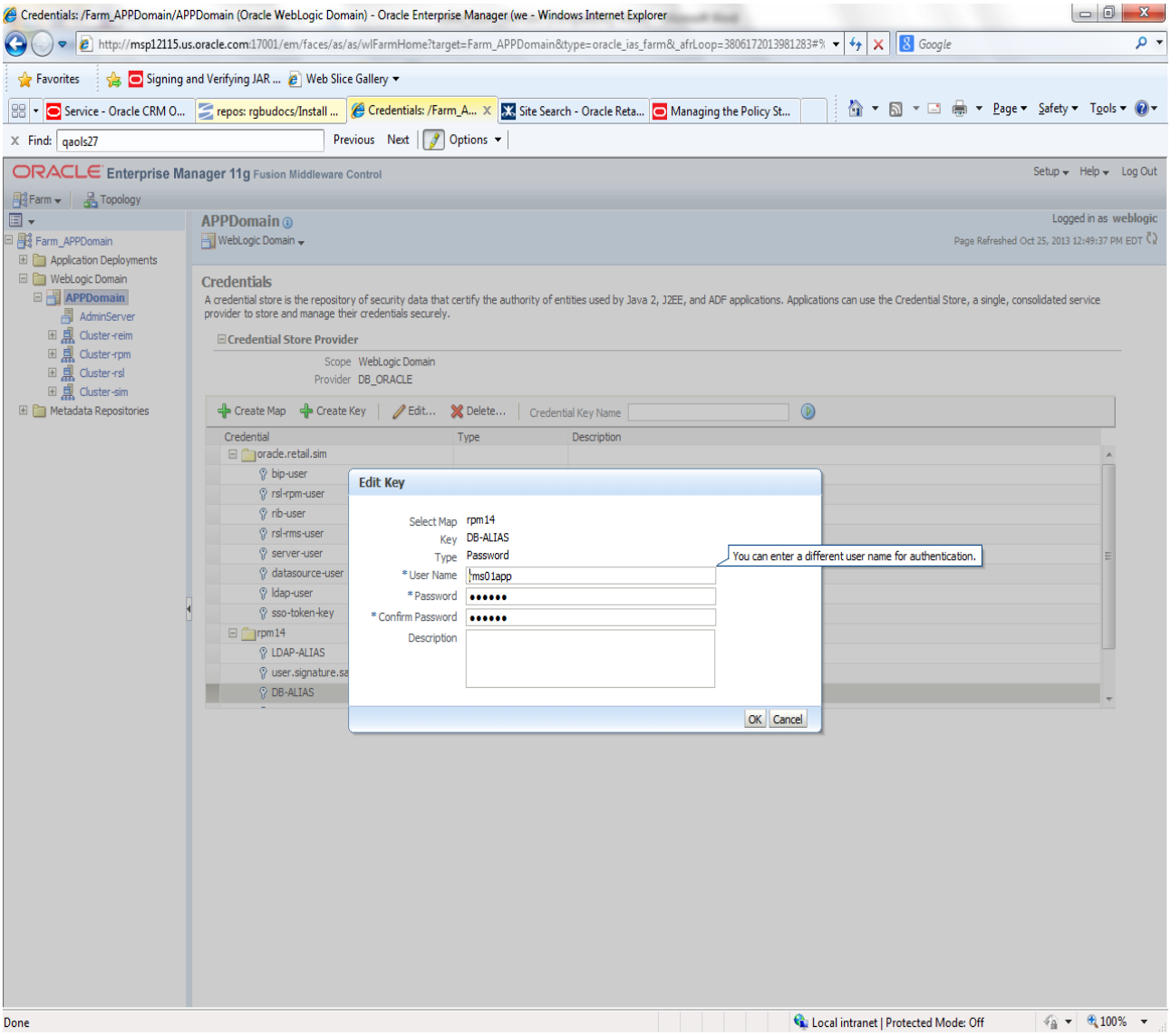


2. Click on Credentials and you will get a screen similar to the following. The following screen is expanded to make it make more sense. From here you can administer credentials.


Credentials: /Farm_APPDomain/APPDomain (Oracle WebLogic Domain) - Oracle Enterprise Manager (we - Windows Internet Explorer
ORACLE Enterprise Manager 11g Fusion Middleware Control
Farm
Topology
Farm_APPDomain
Application Deployments
WebLogic Domain
APPDomain
AdminServer
Cluster rcim
Cluster rpm
Cluster rsl
Cluster sim
Metadata Repositories
APPDomain
WebLogic Domain
Logged in as weblogic
Credentials
A credential store is the repository of security data that certify the authority of entities used by Java 2, J2EE, and ADF applications. Applications can use the Credential Store, a single, consolidated service provider to store and manage their credentials securely.
Credential Store Provider
Scope WebLogic Domain
Provider DB_ORACLE
Create Map
Create Key
Edit...
Delete...
Credential Key Name
Credential
Type
Description
oracle.retail.sim
bip-user
rsl-rpm-user
rib-user
rsl rms user
server-user
datasource-user
ldap user
sso-token-key
rpm14
LDAP-ALIAS
user.signature.salt
DB-ALIAS
Password
Generic
Done
Local intranet | Protected Mode: Off
100%

The Create Map add above is to create a new map with keys under it. A map would usually be an application such as rpm14. The keys will usually represent alias to various users (database user, WebLogic user, LDAP user, etc). The application installer should add the maps so you should not often have to add a map.

Creation of the main keys for an application will also be built by the application installer. You will not be adding keys often as the installer puts the keys out and the keys talk to the application. You may be using EDIT on a key to see what user the key/alias points to and possibly change/reset its password. To edit a key/alias, highlight the key/alias in question and push the edit icon nearer the top of the page. You will then get a screen as follows:



The screen above shows the map (rpm14) that came from the application installer, the key (DB-ALIAS) that came from the application installer (some of the keys/alias are selected by the person who did the application install, some are hard coded by the application installer in question), the type (in this case password), and the user name and password. This is where you would check to see that the user name is correct and reset the password if needed. REMEMBER, a change to an item like a database password

WILL make you come into this and also change the password. Otherwise your application will NOT work correctly.

Managing Credentials with WSLT/OPSS Scripts

This procedure is optional as you can administer the credential store through the Oracle enterprise manager associated with the domain of your application install for RPM, SIM, or Allocation.

An Oracle Platform Security Scripts (OPSS) script is a WLST script, in the context of the Oracle WebLogic Server. An online script is a script that requires a connection to a running server. Unless otherwise stated, scripts listed in this section are online scripts and operate on a database credential store. There are a few scripts that are offline, that is, they do not require a server to be running to operate.

Read-only scripts can be performed only by users in the following WebLogic groups: Monitor, Operator, Configurator, or Admin. Read-write scripts can be performed only by users in the following WebLogic groups: Admin or Configurator. All WLST scripts are available out-of-the-box with the installation of the Oracle WebLogic Server.

WLST scripts can be run in interactive mode or in script mode. In interactive mode, you enter the script at a command-line prompt and view the response immediately after. In script mode, you write scripts in a text file (with a py file name extension) and run it without requiring input, much like the directives in a shell script.

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

The weakness with the WLST/OPSS scripts is that you have to already know your map name and key name. In many cases, you do not know or remember that. The database credential store way through enterprise manager is a better way to find your map and key names easily when you do not already know them. A way in a command line mode to find the map name and alias is to run orapki. An example of orapki is as follows:

msp12115:[1036_APP] /u00/webadmin/product/wls_apps/oracle_common/bin>
./orapki wallet display –wallet
/u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig

(where the path above is the domain location of the wallet)

Output of orapki is below. This shows map name of rpm14 and each alias in the wallet:

Oracle PKI Tool : Version 11.1.1.7.0

Copyright (c) 2004, 2011, Oracle and/or its affiliates. All rights reserved.

Requested Certificates:

User Certificates:

Oracle Secret Store entries:

rpm14@#3#@DB-ALIAS

rpm14@#3#@LDAP-ALIAS

rpm14@#3#@RETAIL.USER

rpm14@#3#@user.signature.salt

rpm14@#3#@user.signature.secretkey

rpm14@#3#@WEBLOGIC-ALIAS

rpm14@#3#@WLS-ALIAS

Trusted Certificates:

Subject: OU=Class 1 Public Primary Certification Authority,O=VeriSign\, Inc.,C=US

OPSS provides the following scripts on all supported platforms to administer credentials (all scripts are online, unless otherwise stated. You need the map name and the key name to run the scripts below

- listCred
- updateCred
- createCred
- deleteCred
- modifyBootStrapCredential
- addBootStrapCredential

listCred

The script `listCred` returns the list of attribute values of a credential in the credential store with given map name and key name. This script lists the data encapsulated in credentials of type password only.

Script Mode Syntax

```
listCred.py -map mapName -key keyName
```

Interactive Mode Syntax

```
listCred(map="mapName", key="keyName")
```

The meanings of the arguments (all required) are as follows:

- `map` specifies a map name (folder).
- `key` specifies a key name.

Examples of Use:

The following invocation returns all the information (such as user name, password, and description) in the credential with map name `myMap` and key name `myKey`:

```
listCred.py -map myMap -key myKey
```

The following example shows how to run this command and similar credential commands with WLST:

```
msp12115:[1036_APP] /u00/webadmin/product/wls_apps/oracle_common/common/bin>
sh wlst.sh

Initializing WebLogic Scripting Tool (WLST)...

Welcome to WebLogic Server Administration Scripting Shell


wls:/offline> connect('weblogic','password123','msp12115.us.oracle.com:17001')
Connecting to t3://msp12115.us.oracle.com:17001 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain
'APPDomain'.

wls:/APPDomain/serverConfig> listCred(map="rpm14",key="DB-ALIAS")
Already in Domain Runtime Tree

[Name : rms01app, Description : null, expiry Date : null]
```

```
PASSWORD:retail
*The above means for map rpm14 in APPDomain, alias DB-ALIAS points to database
user rms01app with a password of retail
```

updateCred

The script `updateCred` modifies the type, user name, and password of a credential in the credential store with given map name and key name. This script updates the data encapsulated in credentials of type password only. Only the interactive mode is supported.

Interactive Mode Syntax

```
updateCred(map="mapName", key="keyName", user="userName",
password="passW", [desc="description"])
```

The meanings of the arguments (optional arguments are enclosed by square brackets) are as follows:

- `map` specifies a map name (folder) in the credential store.
- `key` specifies a key name.
- `user` specifies the credential user name.
- `password` specifies the credential password.
- `desc` specifies a string describing the credential.

Example of Use:

The following invocation updates the user name, password, and description of the password credential with map name `myMap` and key name `myKey`:

```
updateCred(map="myMap", key="myKey", user="myUsr",
password="myPassw")
```

createCred

The script `createCred` creates a credential in the credential store with a given map name, key name, user name and password. This script can create a credential of type password only. Only the interactive mode is supported.

Interactive Mode Syntax

```
createCred(map="mapName", key="keyName", user="userName", password="passW",
[desc="description"])
```

The meanings of the arguments (optional arguments are enclosed by square brackets) are as follows:

- `map` specifies the map name (folder) of the credential.
- `key` specifies the key name of the credential.
- `user` specifies the credential user name.
- `password` specifies the credential password.
- `desc` specifies a string describing the credential.

Example of Use:

The following invocation creates a password credential with the specified data:

```
createCred(map="myMap", key="myKey", user="myUsr", password="myPassw")
```

deleteCred

The script `deleteCred` removes a credential with given map name and key name from the credential store.

Script Mode Syntax

```
deleteCred.py -map mapName -key keyName
```

Interactive Mode Syntax

```
deleteCred(map="mapName",key="keyName")
```

The meanings of the arguments (all required) are as follows:

- `map` specifies a map name (folder).
- `key` specifies a key name.

Example of Use:

The following invocation removes the credential with map name `myMap` and key name `myKey`:

```
deleteCred.py -map myMap -key myKey
```

modifyBootStrapCredential

The offline script `modifyBootStrapCredential` modifies the bootstrap credentials configured in the default jps context, and it is typically used in the following scenario: suppose that the policy and credential stores are LDAP-based, and the credentials to access the LDAP store (stored in the LDAP server) are changed. Then this script can be used to seed those changes into the bootstrap credential store.

This script is available in interactive mode only.

Interactive Mode Syntax

```
modifyBootStrapCredential(jpsConfigFile="pathName",
username="usrName", password="usrPass")
```

The meanings of the arguments (all required) are as follows:

- `jpsConfigFile` specifies the location of the file `jps-config.xml` relative to the location where the script is run. Example location: /u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig. Example location of the bootstrap wallet is /u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig/bootstrap
- `username` specifies the distinguished name of the user in the LDAP store.
- `password` specifies the password of the user.

Example of Use:

Suppose that in the LDAP store, the password of the user with distinguished name `cn=orcladmin` has been changed to `welcome1`, and that the configuration file `jps-config.xml` is located in the current directory.Then the following invocation changes the password in the bootstrap credential store to `welcome1`:

```
modifyBootStrapCredential(jpsConfigFile='./jps-config.xml',
username='cn=orcladmin', password='welcome1')
```

Any output regarding the audit service can be disregarded.

addBootStrapCredential

The offline script `addBootStrapCredential` adds a password credential with given map, key, user name, and user password to the bootstrap credentials configured in the default jps context of a jps configuration file.

Classloaders contain a hierarchy with parent classloaders and child classloaders. The relationship between parent and child classloaders is analogous to the object relationship of super classes and subclasses. The bootstrap classloader is the root of the Java classloader hierarchy. The Java virtual machine (JVM) creates the bootstrap classloader, which loads the Java development kit (JDK) internal classes and `java.*` packages included in the JVM. (For example, the bootstrap classloader loads `java.lang.String`.)

This script is available in interactive mode only.

Interactive Mode Syntax

```
addBootStrapCredential(jpsConfigFile="pathName", map="mapName",
key="keyName", username="usrName", password="usrPass")
```

The meanings of the arguments (all required) are as follows:

- `jpsConfigFile` specifies the location of the file `jps-config.xml` relative to the location where the script is run. Example location: /u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig
- `map` specifies the map of the credential to add.
- `key` specifies the key of the credential to add.
- `username` specifies the name of the user in the credential to add.
- `password` specifies the password of the user in the credential to add.

Example of Use:

The following invocation adds a credential to the bootstrap credential store:

```
addBootStrapCredential(jpsConfigFile='./jps-config.xml', map='myMapName',
key='myKeyName', username='myUser', password ='myPass')
```

Quick Guide for Retail Password Stores (db wallet, java wallet, DB credential stores)

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
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	RPM plsql and sqlldr batches
RWMS auto-login	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	<RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms14inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	n/a	Alias hard-coded by installer

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip14	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip14	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip14	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	DB credential store		Map=rpm14 or what you called the app at install time.	Many for app use					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file.
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm14	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm14	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			rpm14	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	
			rpm14	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm14	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm14	<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, ex: reim14>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name, ex: reim14>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name, ex: reim14>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name, ex: reim14>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
Alloc app	DB credential store		Map=alloc 14 or what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file. The bootstrap directory under this directory has bootstrap cwallet.sso file.
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/config							Each alias must be unique
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms schema 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	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc batch user alias>	<SYSTEM_ ADMINISTRATOR>	Batch use	Installer	alloc14	
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	DB credential store		Map=oracle.retail.sim	Aliases required for SIM app use					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file.
	JAVA	<weblogic domain home>/retail/<deployed sim app name>/config	<installed sim app name>	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			<installed sim app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			<installed sim app name>	<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	
RFI	Java	<RFI-HOME>/retail-financial-integration-solution/service-based-integration/conf/security							

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	rfiAppServerAdminServerUserAlias	<rfi weblogic user name>	App use	Installer	rfiAppServerAdminServerUserAlias	
			<installed app name>	rfiAdminUiUserAlias	<ORFI admin user>	App use	Installer	rfiAdminUiUserAlias	
			<installed app name>	rfiDataSourceUserAlias	<ORFI schema user name>	App use	Installer	rfiDataSourceUserAlias	
			<installed app name>	ebsDataSourceUserAlias	<EBS schema user name>	App use	Installer	ebsDataSourceUserAlias	
			<installed app name>	smtpMailFromAddressAlias	<From email address>	App use	Installer	smtpMailFromAddressAlias	

E

Appendix: Single Sign-On for WebLogic

Single Sign-On (SSO) is a term for the ability to sign onto multiple Web applications via a single user ID/Password. There are many implementations of SSO. Oracle provides an implementation with Oracle Access Manager.

Most, if not all, SSO technologies use a session cookie to hold encrypted data passed to each application. The SSO infrastructure has the responsibility to validate these cookies and, possibly, update this information. The user is directed to log on only if the cookie is not present or has become invalid. These session cookies are restricted to a single browser session and are never written to a file.

Another facet of SSO is how these technologies redirect a user's Web browser to various servlets. The SSO implementation determines when and where these redirects occur and what the final screen shown to the user is.

Most SSO implementations are performed in an application's infrastructure and not in the application logic itself. Applications that leverage infrastructure managed authentication (such as deployment specifying Basic or Form authentication) typically have little or no code changes when adapted to work in an SSO environment.

What Do I Need for Single Sign-On?

A Single Sign-On system involves the integration of several components, including Oracle Identity Management and Oracle Access Management. This includes the following components:

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle Access Manager (OAM) 11g Release 1 server and administrative console for implementing and configuring policies for single sign-on.
- A Policy Enforcement Agent such as Oracle Access Manager 11g Agent (WebGate), used to authenticate the user and create the Single Sign-On cookies.
- Oracle Directory Services Manager (ODSM) application in OIM11g, used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OAM system and registering HTTP servers.

Additional WebLogic managed servers will be needed to deploy the business applications leveraging the Single Sign-On technology.

Can Oracle Access Manager Work with Other SSO Implementations?

Yes, Oracle Access Manager has the ability to interoperate with many other SSO implementations, but some restrictions exist.

Oracle Single Sign-on Terms and Definitions

The following terms apply to single sign-on.

Authentication

Authentication is the process of establishing a user's identity. There are many types of authentication. The most common authentication process involves a user ID and password.

Dynamically Protected URLs

A Dynamically Protected URL is a URL whose implementing application is aware of the Oracle Access Manager environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic protection typically display a Login link to provide user authentication and gain greater access to the application's resources.

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

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

MOD_WEBLOGIC

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

Oracle Access Manager 11g Agent (WebGate)

Oracle WebGates are policy enforcement agents which reside with relying parties and delegate authentication and authorization tasks to OAM servers.

Oracle Internet Directory

Oracle Internet Directory (OID) is an LDAP-compliant directory service. It contains user ids, passwords, group membership, privileges, and other attributes for users who are authenticated using Oracle Access Manager.

Partner Application

A partner application is an application that delegates authentication to the Oracle Identity Management Infrastructure. One such partner application is the Oracle HTTP Server (OHS) supplied with Oracle Forms Server or WebTier11g Server if using other Retail Applications other than Oracle Forms Applications.

All partner applications must be registered with Oracle Access Manager (OAM) 11g. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Statically Protected URLs

A URL is considered to be Statically Protected when an Oracle HTTP server is configured to limit access to this URL to only SSO authenticated users. Any unauthenticated attempt to access a Statically Protected URL results in the display of a login page or an error page to the user.

Servlets, static HTML pages, and JSP pages may be statically protected.

What Single Sign-On is not

Single Sign-On is NOT a user ID/password mapping technology.

However, some applications can store and retrieve user IDs and passwords for non-SSO applications within an OID LDAP server. An example of this is the Oracle Forms Web Application framework, which maps Single Sign-On user IDs to a database logins on a per-application basis.

How Oracle Single Sign-On Works

Oracle Access Manager involves several different components. These are:

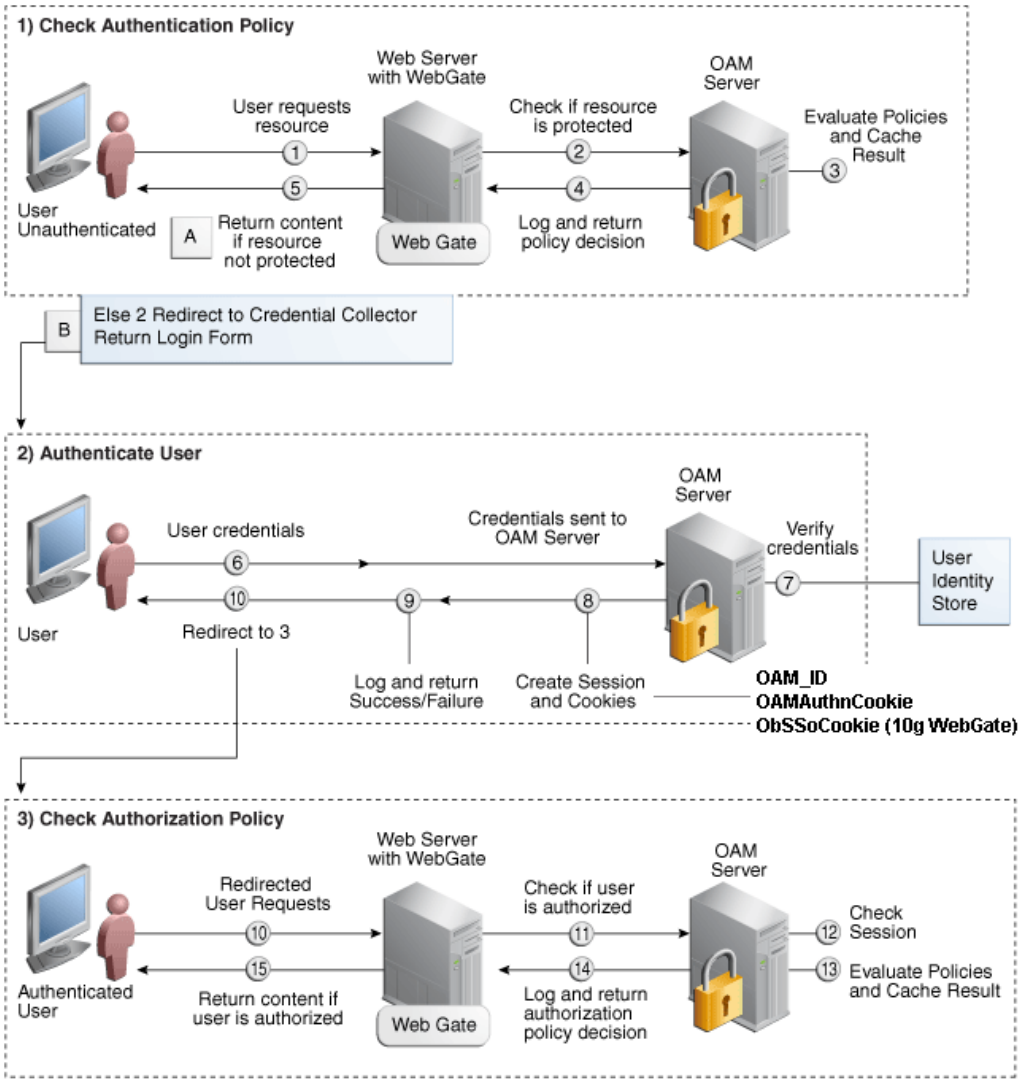
- The Oracle Access Manager (OAM) server, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle Access Manager Agent associated with the Web application, which verifies and controls browser redirection to the Oracle Access Manager server.
- If the Web application implements dynamic protection, then the Web application itself is involved with the OAM system.

About SSO Login Processing with OAM Agents

1. The user requests a resource.
2. Webgate forwards the request to OAM for policy evaluation
3. OAM:
   a. Checks for the existence of an SSO cookie.
   b. Checks policies to determine if the resource is protected and if so, how?
4. OAM Server logs and returns the decision
5. Webgate responds as follows:
   - **Unprotected Resource:** Resource is served to the user
   - **Protected Resource:**
     Resource is redirected to the credential collector.
     The login form is served based on the authentication policy.
     Authentication processing begins
6. User sends credentials
7. OAM verifies credentials
8. OAM starts the session and creates the following host-based cookies:
   - **One per partner:** OAMAuthnCookie set by 11g WebGates using authentication token received from the OAM Server after successful authentication.
     Note: A valid cookie is required for a session.
   - **One for OAM Server:** OAM_ID
9. OAM logs Success of Failure.
10. Credential collector redirects to WebGate and authorization processing begins.
11. WebGate prompts OAM to look up policies, compare them to the user's identity, and determine the user's level of authorization.
12. OAM logs policy decision and checks the session cookie.
13. OAM Server evaluates authorization policies and cache the result.
14. OAM Server logs and returns decisions
15. WebGate responds as follows:
    - If the authorization policy allows access, the desired content or applications are served to the user.

- If the authorization policy denies access, the user is redirected to another URL determined by the administrator.

SSO Login Processing with OAM Agents



Installation Overview

Installing an Oracle Retail supported Single Sign-On installation using OAM11g requires installation of the following:

1. Oracle Internet Directory (OID) LDAP server and the Oracle Directory Services Manager. They are typically installed using the Installer of Oracle Identity Management 11gR1 (11.1.1.7). The ODSM application can be used for user and realm management within OID.
2. Oracle Access Manager 11gR1 (11.1.1.7) 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.

Infrastructure Installation and Configuration

The Infrastructure installation for Oracle Access Manager (OAM) is dependent on the environment and requirements for its use. Deploying Oracle Access Manager (OAM) to be used in a test environment does not have the same availability requirements as for a production environment. Similarly, the Oracle Internet Directory (OID) LDAP server can be deployed in a variety of different configurations. See the *Oracle Identity Management Installation Guide11g*.

OID User Data

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

Customers with existing corporate LDAP implementations may need to synchronize user information between their existing LDAP directory servers and OID. OID supports standard LDIF file formats and provides a JNDI compliant set of Java classes as well. Moreover, OID provides additional synchronization and replication facilities to integrate with other corporate LDAP implementations.

Each user ID stored in OID has a specific record containing user specific information. For role-based access, groups of users can be defined and managed within OID. Applications can thus grant access based on group (role) membership saving administration time and providing a more secure implementation.

User Management

User Management consists of displaying, creating, updating or removing user information. There are many methods of managing an LDAP directory including LDIF scripts or Oracle Directory Services Manager (ODSM) available for OID11g.

ODSM

Oracle Directory Services Manager (ODSM) is a Web-based application used in OID11g is designed for both administrators and users which enables you to configure the structure of the directory, define objects in the directory, add and configure users, groups, and other entries. ODSM is the interface you use to manage entries, schema, security, adapters, extensions, and other directory features.

LDIF Scripts

Script based user management can be used to synchronize data between multiple LDAP servers. The standard format for these scripts is the LDAP Data Interchange Format (LDIF). OID supports LDIF script for importing and exporting user information. LDIF scripts may also be used for bulk user load operations.

User Data Synchronization

The user store for Oracle Access Manager resides within the Oracle Internet Directory (OID) LDAP server. Oracle Retail applications may require additional information attached to a user name for application-specific purposes and may be stored in an application-specific database. Currently, there are no Oracle Retail tools for synchronizing changes in OID stored information with application-specific user stores. Implementers should plan appropriate time and resources for this process. Oracle Retail

strongly suggests that you configure any Oracle Retail application using an LDAP for its user store to point to the same OID server used with Oracle Access Manager.

F

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)
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 Modeling Engine (ORME)
17. Oracle Retail Assortment Space Optimization (OASO)

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