Oracle® Retail Invoice Matching

Installation Guide
Release 13.2.3
E25281-02

November 2011

ORACLE

Oracle® Retail Invoice Matching Installation Guide, Release 13.2.3

Primary Author: Wade Schwarz

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 Invoice Matching Installation Guide, Release 13.2.3

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 Invoice Matching Release 13.2.3 documentation set:

- *Oracle Retail Invoice Matching Release Notes*
- *Oracle Retail Invoice Matching Installation Guide*
- *Oracle Retail Invoice Matching Operations Guide*
- *Oracle Retail Invoice Matching User Guide*
- *Oracle Retail Invoice Matching Online Help*
- *Oracle Retail Invoice Matching Data Model*
- *Oracle Retail Merchandising Implementation 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, 13.2) or a later patch release (for example, 13.2.1). If you are installing the base release and additional patch and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

Conventions

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

```
This is a code sample
    It is used to display examples of code
```

1

Preinstallation Tasks

This chapter explains the tasks required prior to installation.

Check for the Current Version of the Installation Guide

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

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

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

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

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

Check Supported Database Server Requirements

General requirements for a database server running Oracle Retail Invoice Matching include:

Supported on	Versions Supported
Database Server OS	OS certified with Oracle Database 11gR2 Enterprise Edition. Options are: ▪ Oracle Linux 5 Update 5 for x86-64 (actual hardware or Oracle virtual machine) ▪ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.5) for x86-64 (actual hardware or Oracle virtual machine) ▪ IBM AIX 6.1 (actual hardware or LPARs) ▪ Solaris 10 Sparc (actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.2) with the following specifications: **Components:** ▪ Oracle Partitioning ▪ Examples CD (Formerly the companion CD) **One-off Patches:** ▪ 10170431 – CTWR consumes a lot of CPU cycles. If ASM is used, apply the following patch to Oracle Database home: ▪ 11808931 – Merge request on top of 11.2.0.2.0 for defects 10410054 and 10422126. **Other components:** ▪ Perl compiler 5.0 or later ▪ X-Windows interface

Check Supported Application Server Requirements

General requirements for an application server capable of running the Oracle Retail Invoice Matching application include the following:

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

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

Supported on	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.3). Options are: ▪ Oracle Linux 5 Update 5 for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.5) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or Logical Domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.3) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.3) **Other components:** ▪ Oracle BI Publisher 10g (10.1.3.4) **Optional (SSO required)** ▪ Oracle Internet Directory 10g (10.1.4) ▪ Oracle WebTier 11g (11.1.1.3) **Note:** See Installer Fails because of missing .jar in $ORACLE_HOME/utils/ccr/lib in Appendix: Common Installation Errors. This issue occurs only when the application is being installed on the same WebLogic server on which forms based applications are installed. It is valid only for Linux 64-bit.

Verify Single Sign-On

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

If Single Sign-On is to be used, verify the Oracle Infrastructure Server 10g has been installed. Verify the Oracle WebTier Server is registered with the Infrastructure Oracle Internet Directory as a partner application.

Check Supported Client PC and Web Browser Requirements

Requirement	Version
Operating system	Windows XP
Display resolution	1024x768 or higher
Processor	2.6GHz or higher
Memory	1GByte or higher
Networking	intranet with at least 10Mbps data rate
Sun Java Runtime Environment	1.6.0_22+
Browser	Microsoft Internet Explorer version 7.0

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)/Oracle Retail Trade Management (RTM)/Oracle Retail Sales Audit (ReSA)	13.2.3
SIM (by way of RMS)	13.2.3

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.

Supported Oracle Applications

Requirement	Version
Oracle E-Business Suite (Accounts Payable)	Oracle Application Integration Architecture (AIA) Media Pack 2.5 Oracle E-Business Suite integration is supported using the Oracle Financial Operations Control Integration Pack for Oracle Retail Merchandising Suite and Oracle E-Business Suite Financials. See the *Oracle® Application Integration Architecture 2.5: Installation and Upgrade Guide* for specific version information.
PeopleSoft Enterprise Financials	Oracle Application Integration Architecture (AIA) Media Pack 2.5 PeopleSoft Enterprise Financials integration is supported using the Oracle Retail Merchandising Integration Pack for PeopleSoft Enterprise Financials: Financial Operations Control. See the *Oracle Application Integration Architecture 2.5: Installation and Upgrade Guide* for specific version information.

2

RAC and Clustering

Oracle Retail Invoice Matching 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 11.2.0.2 RAC database. When using a RAC database, all JDBC connections should be configured to use OCI connections rather than THIN connections. It is suggested that when using OCI connections, the Oracle Retail products database be configured in the tnsnames.ora file used by the WebLogic Server installations.

Clustering for WebLogic Server 10.3.3 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 11.2.0.2 Oracle Internet Directory database with the WebLogic 10.3.3 cluster. It is suggested that a Web Tier 11.1.1.3 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

The ReIM database objects are bundled with the RMS database schema installer. To install the ReIM database objects follow the *Oracle Retail Merchandising System Installation Guide* to run the database schema installer, and select the ReIM option on the product selection page.

4

Application Installation Tasks

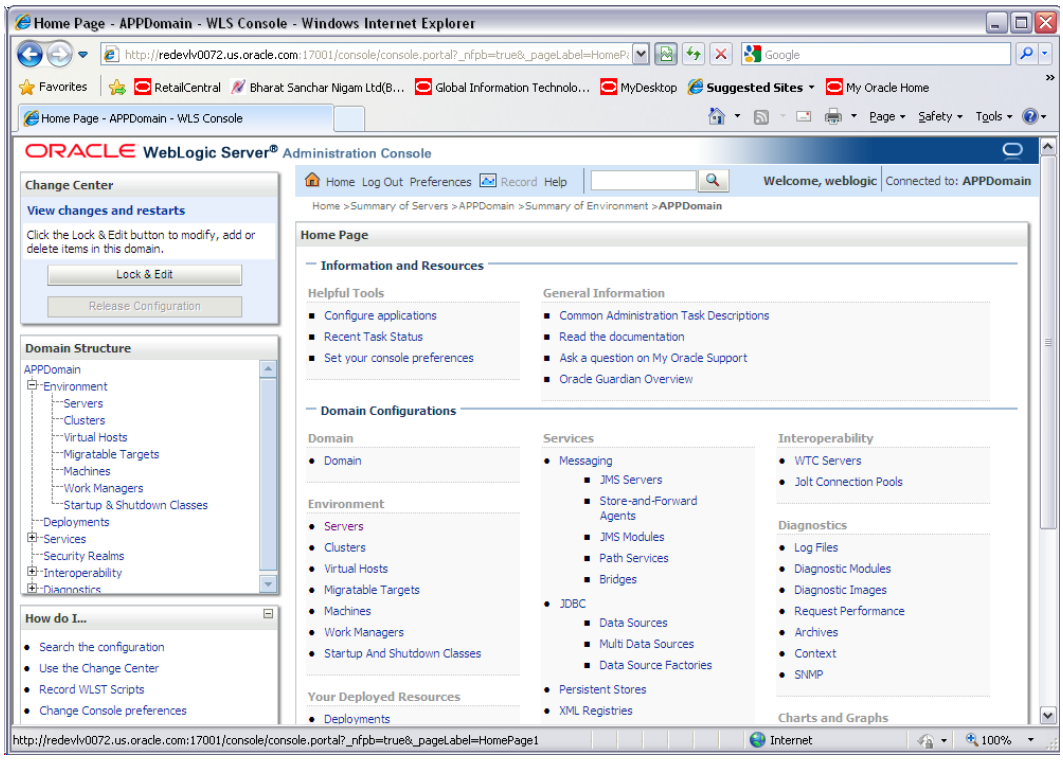
These instructions apply to new installations. Before proceeding, you must install Oracle WebLogic Server 11g Release 1 (10.3.3) and patches listed in the Chapter 1 of this document. The Oracle Retail Invoice Matching application is deployed to a WebLogic Managed server within the Web Logic installation. It is assumed that Oracle Database has already been configured and loaded with the appropriate RMS and Oracle Retail Invoice Matching schemas for your installation.

Install Managed Server in WebLogic

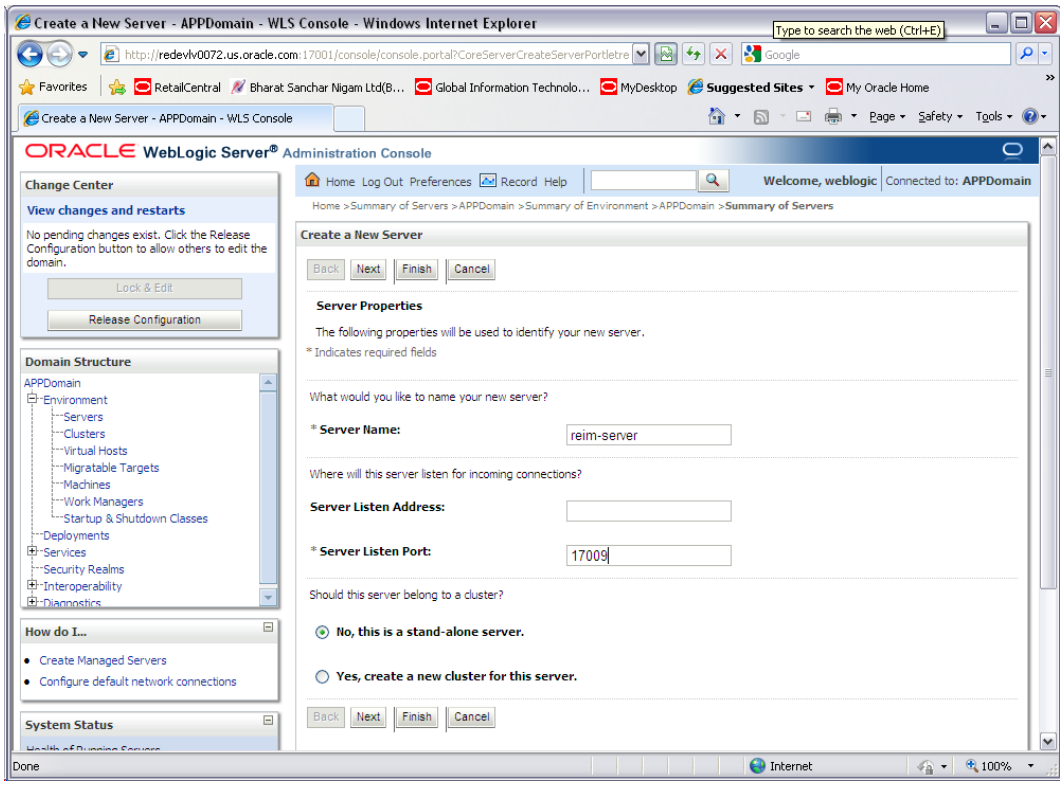
Important Note: Skip this section if a managed server already exists for Oracle Retail Invoice Matching.

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

1. Log in to the Administration Console.



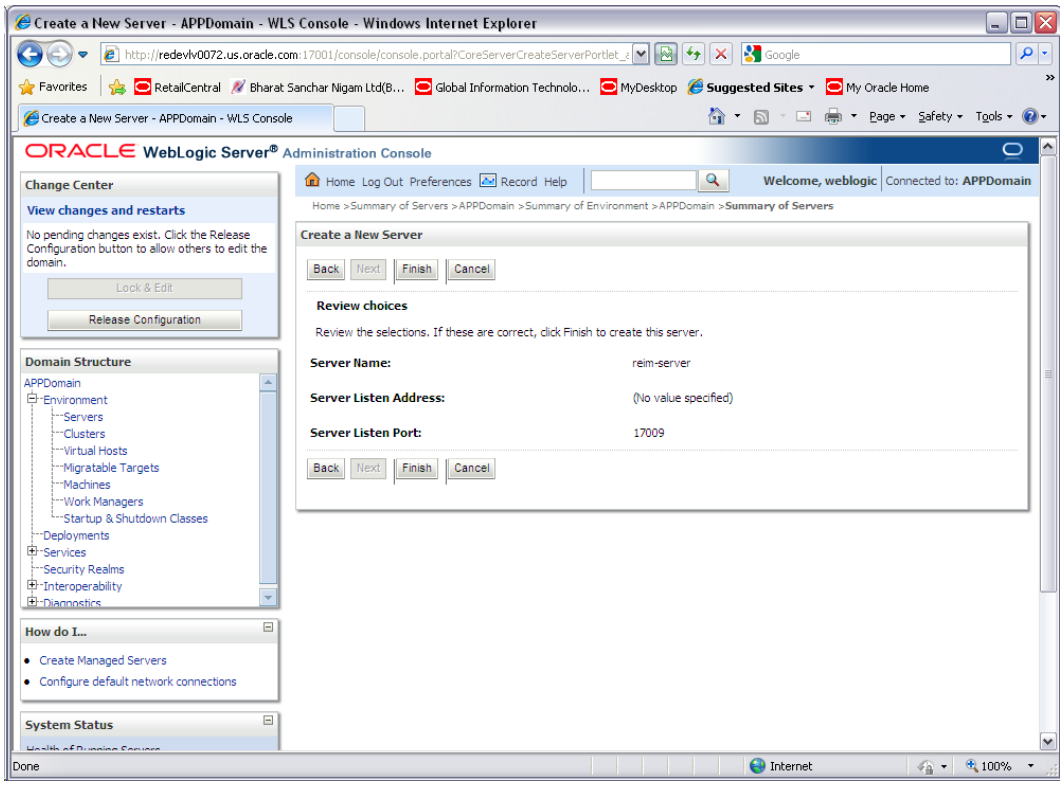
2. Click **Lock & Edit**.
3. Navigate to Environment > Servers and select new tab of the servers on the right side.



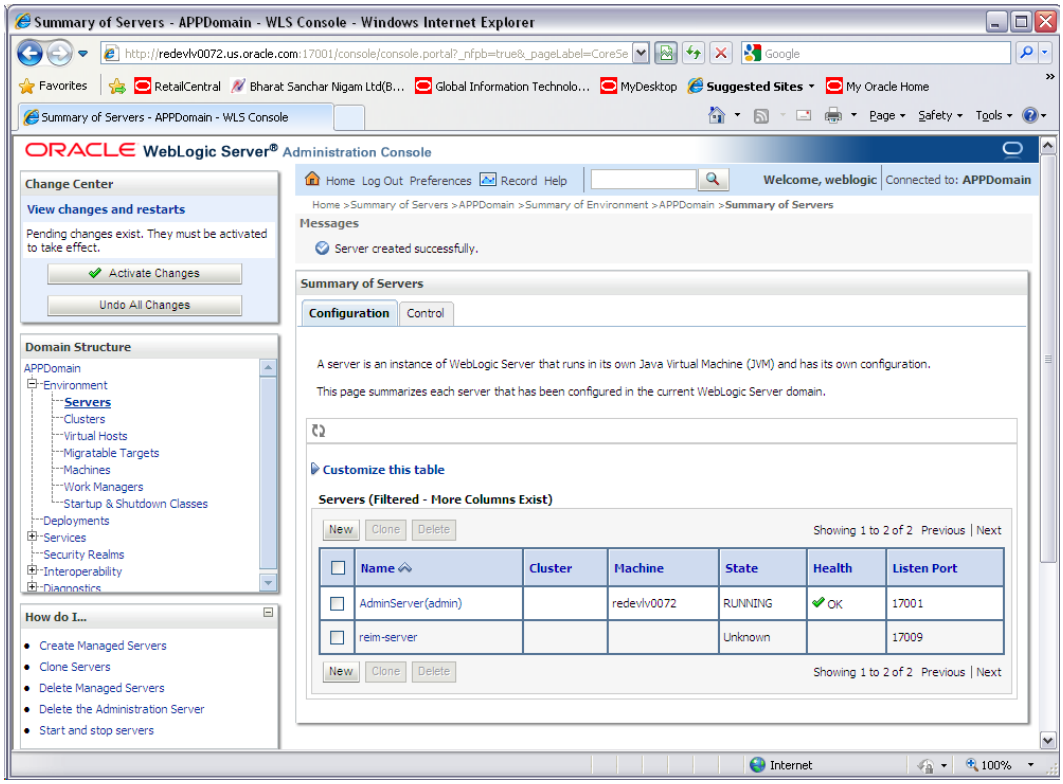
4. Set the following variables:
   - **Server Name**: These should be some name specific to your application targeted (for example, reim-server).

     Server Listen Address: <weblogic server> (for example, redevlv0072.us.oracle.com)
   - **Server Listen Port**: A free port. Check for availability.

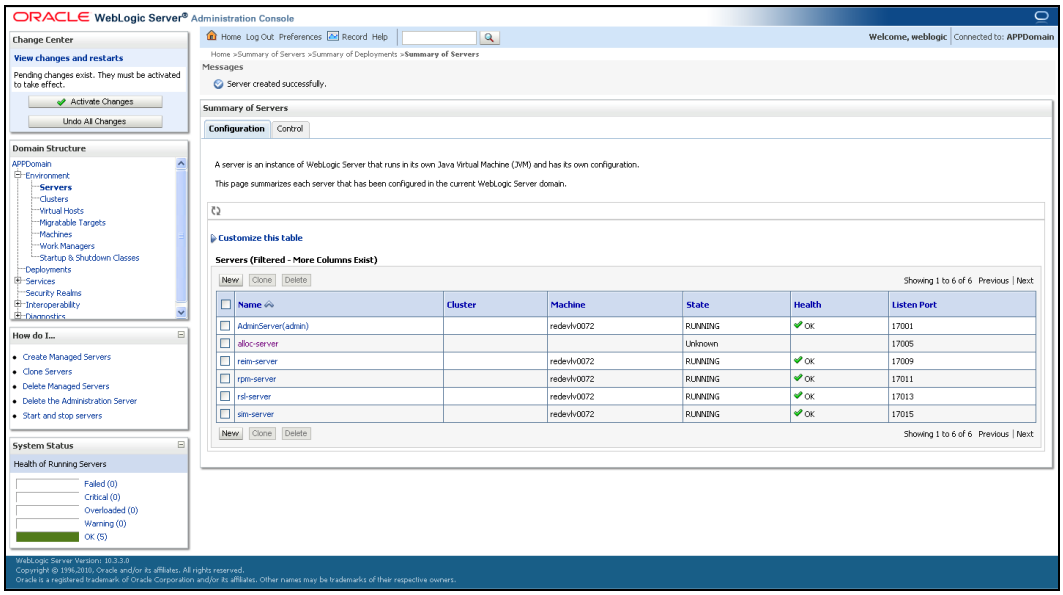
     A suggestion is to increment the AdminServer port by two and keep incrementing by two for each managed server (for example, 17003, 17005, 17007, 17009, and so on.)
5. Click **Next**.



6. Click **Finish**.



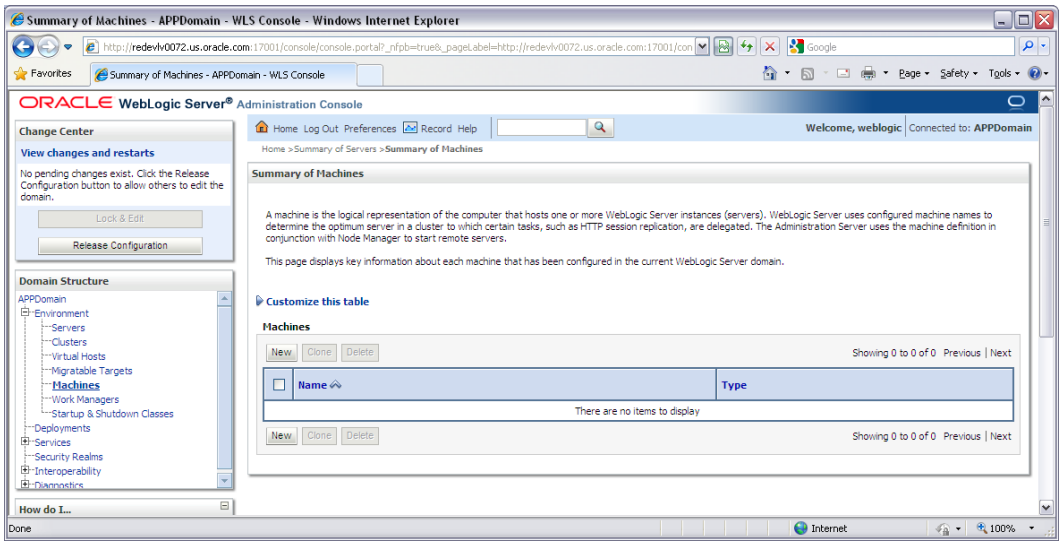
7. Click **Activate Changes** on the left side.



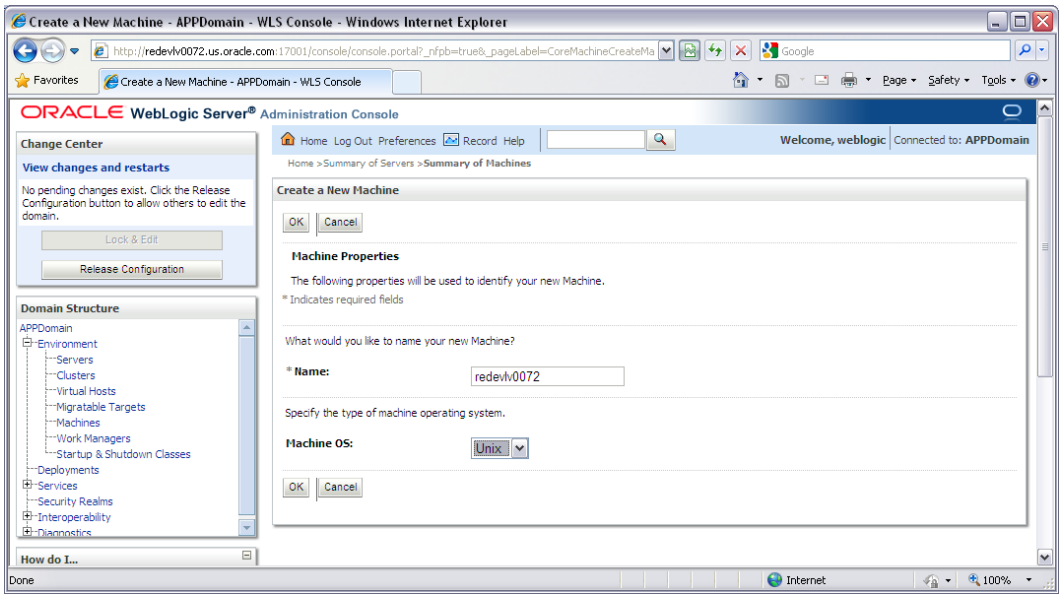
Install Node Manager

Install Node Manager if it was not created during domain install. The node manager is required so that the managed servers can be started and stopped through the Administration Console. Only one node manager is needed per WebLogic install.

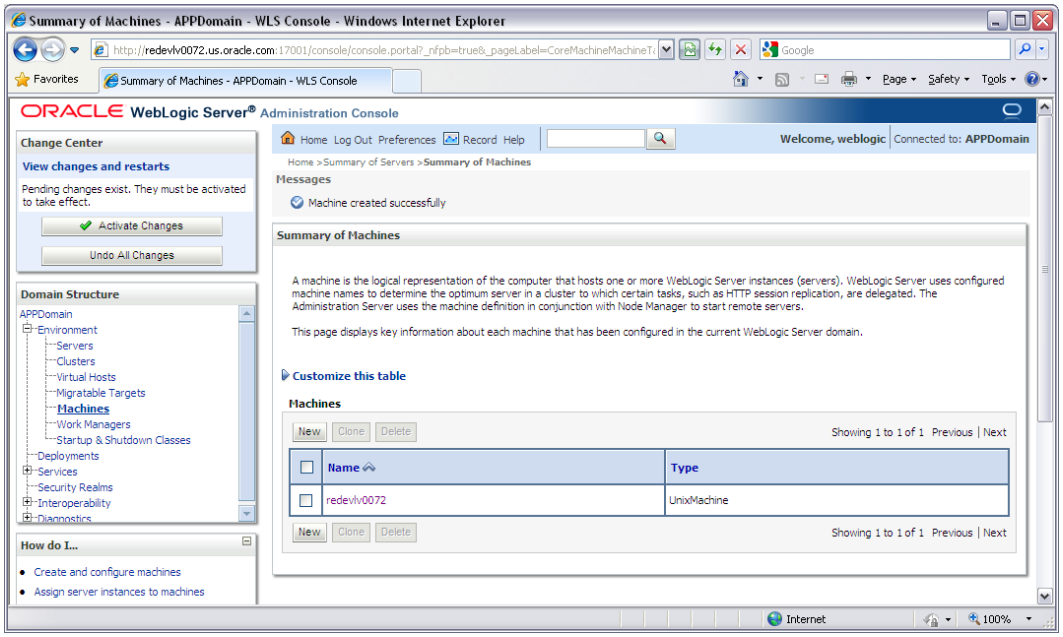
1. Log in to the Administration Console.
2. Click **Lock & Edit** and navigate to Environments > Machines.



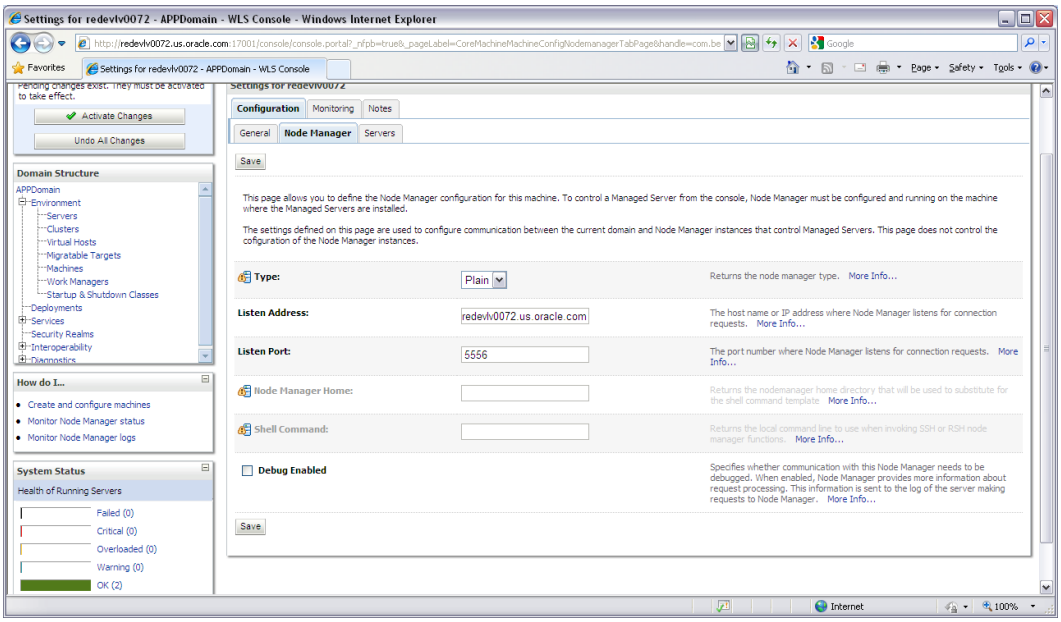
3. Click **New**.



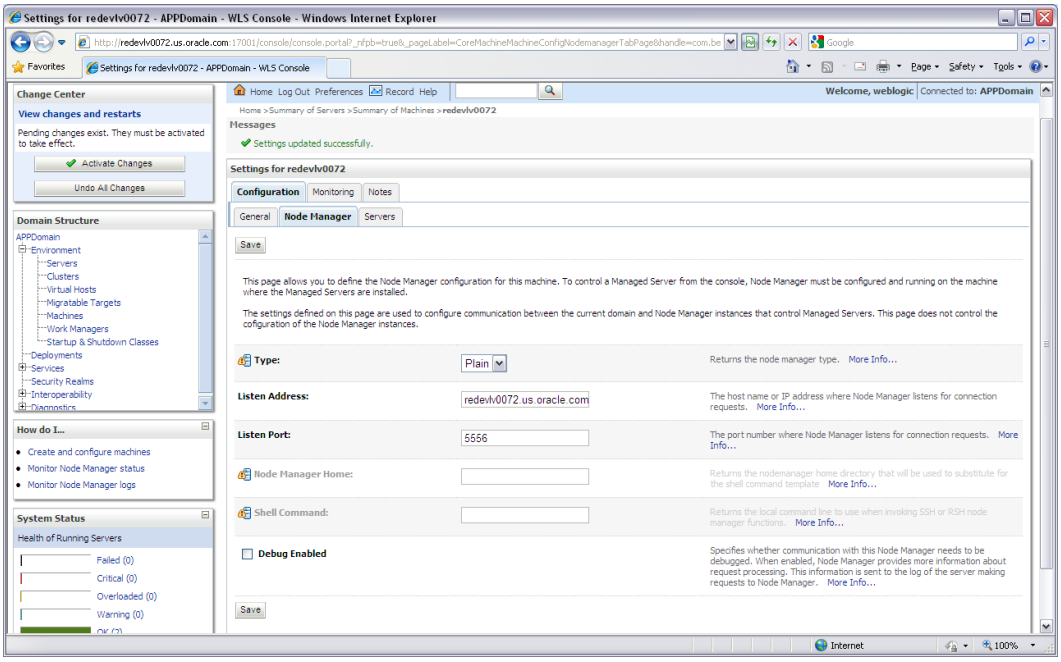
4. Set the following variables.
   - **Name**: Logical machine name
   - **Machine OS**: UNIX
5. Click **OK**.
6. Click on the machine created.



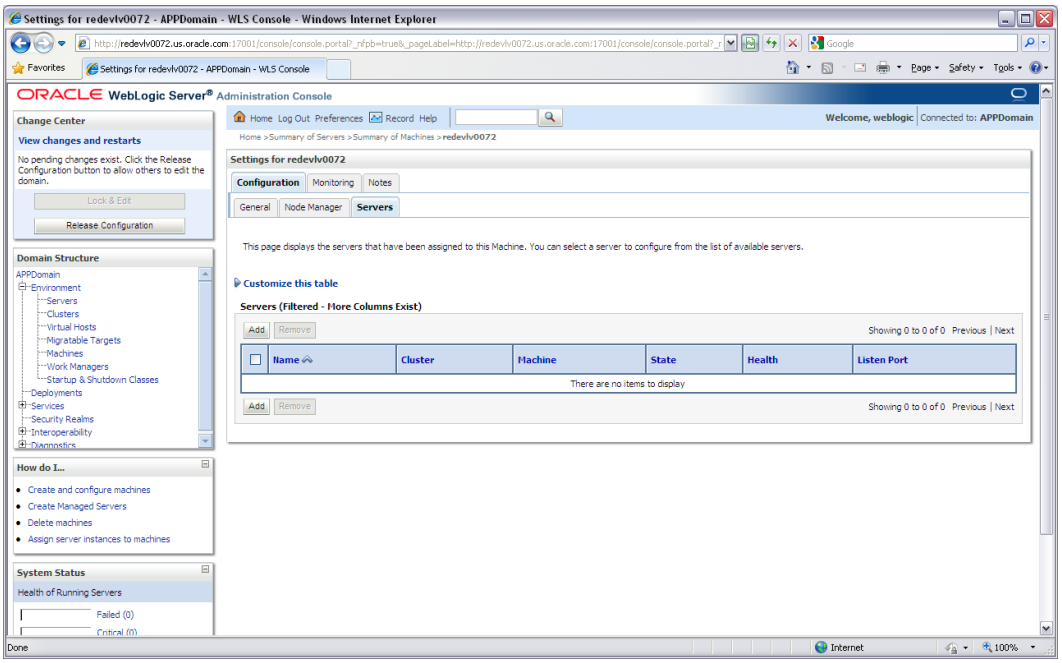
7. Click the Node manager tab and update the details below.
   - **Type**: Plain
   - **Listen Address**: redevlv0072.us.oracle.com
   - **Listen Port**: Default port (for example, 5556) or any available port.



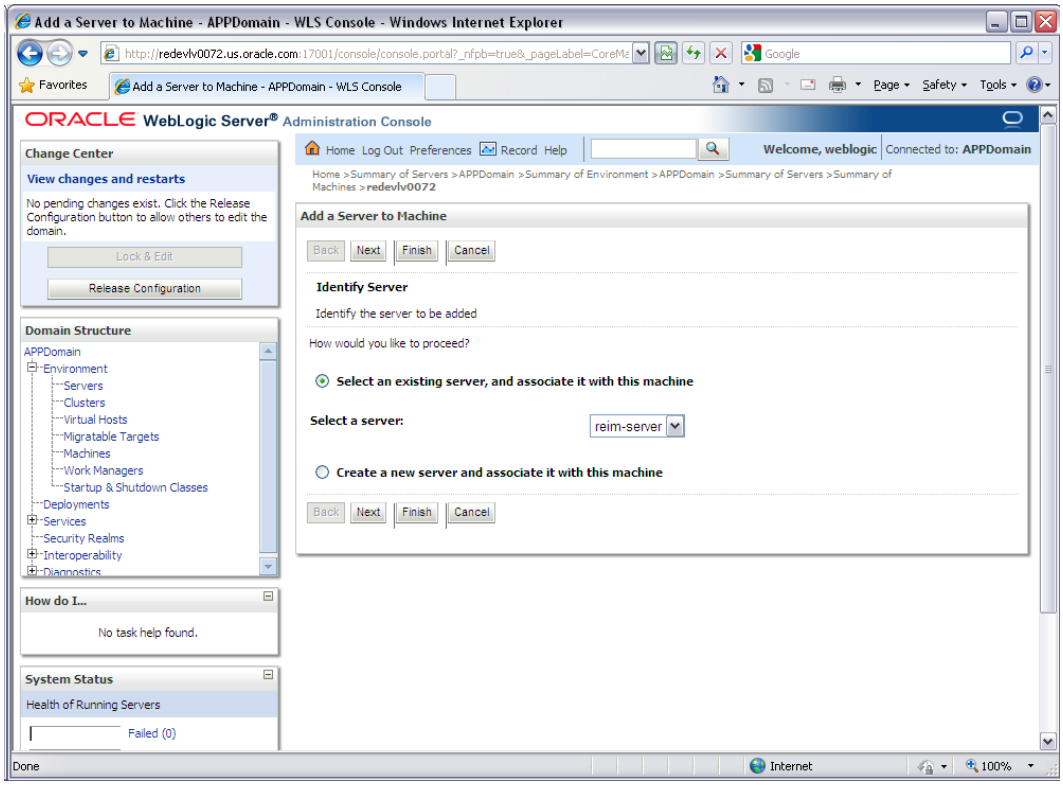
8. Click **Save**.



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



12. Add the managed servers that need to be configured with the Node Manager. Save the changes.
13. Click **Add**.



14. Set the following variables:
    - Server: reim-server
15. Click **Next/Finish**.

16. Click **Activate Changes**.

> **Note:** To activate changes the server needs to be stopped as follows.
>
> $WLS_HOME/user_projects/domains/<APP_Domain>/bi
> n/stopManagedWebLogic.sh reim-server
> ${server_name}:${server_port}
>
> Go to the managed server that is being added to the machine and click the Server Start tab. In the Class Path box, add the following:
> <full-path-to-domain>/servers/<managed-server>
> For example: /u00/webadmin/product/10.3.3
> /WLS/user_projects/domains/<Domain_name>/servers/r
> eim-server

17. Start NodeManager from the server using the startNodeManager.sh at $WLS_HOME/wlserver_10.3/server/bin:
18. Update nodemanager.properties file at the following location and set the SecureListener variable to false:

    $WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties
    SecureListener=false.
19. NodeManager must be restarted after making changes to the nodemanager.properties file.

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

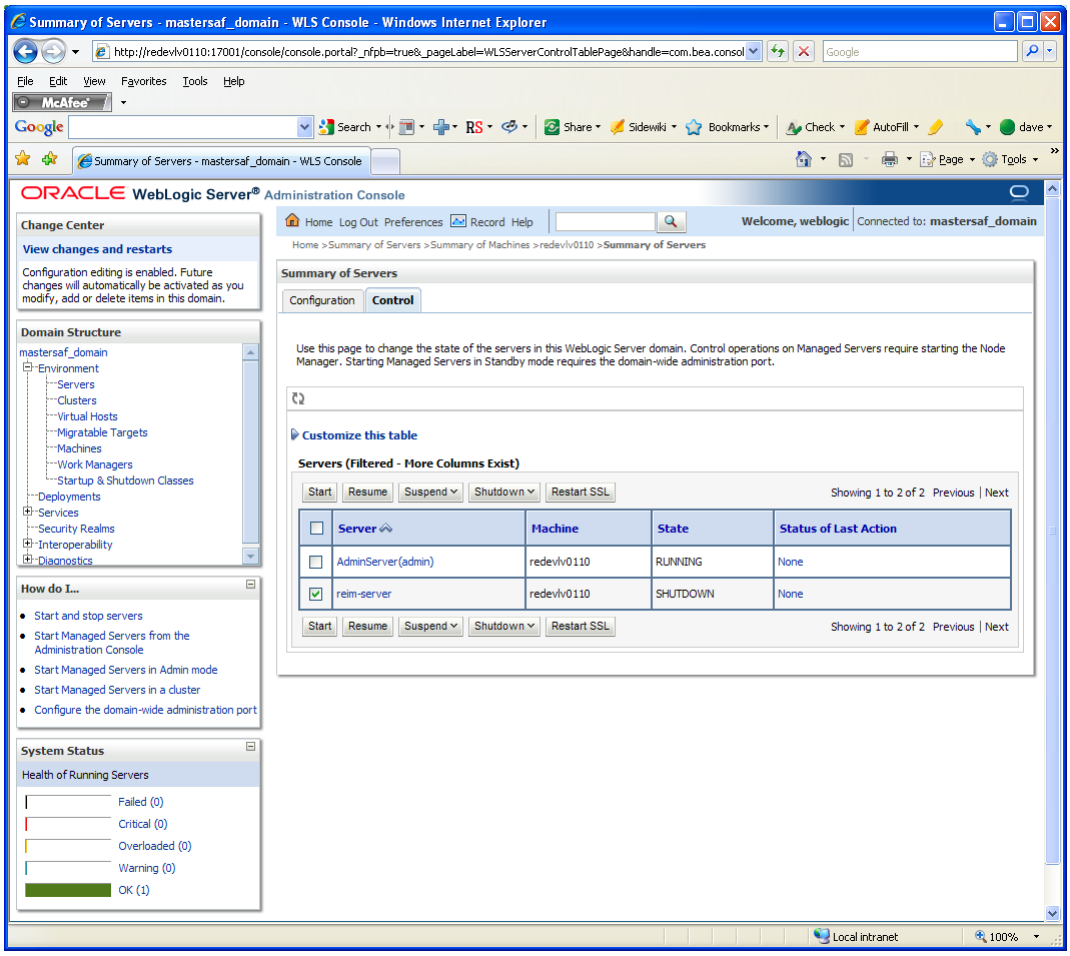
Start the Managed Servers

To start the managed servers, do the following.

1. Start the Node Manager from the command line.

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

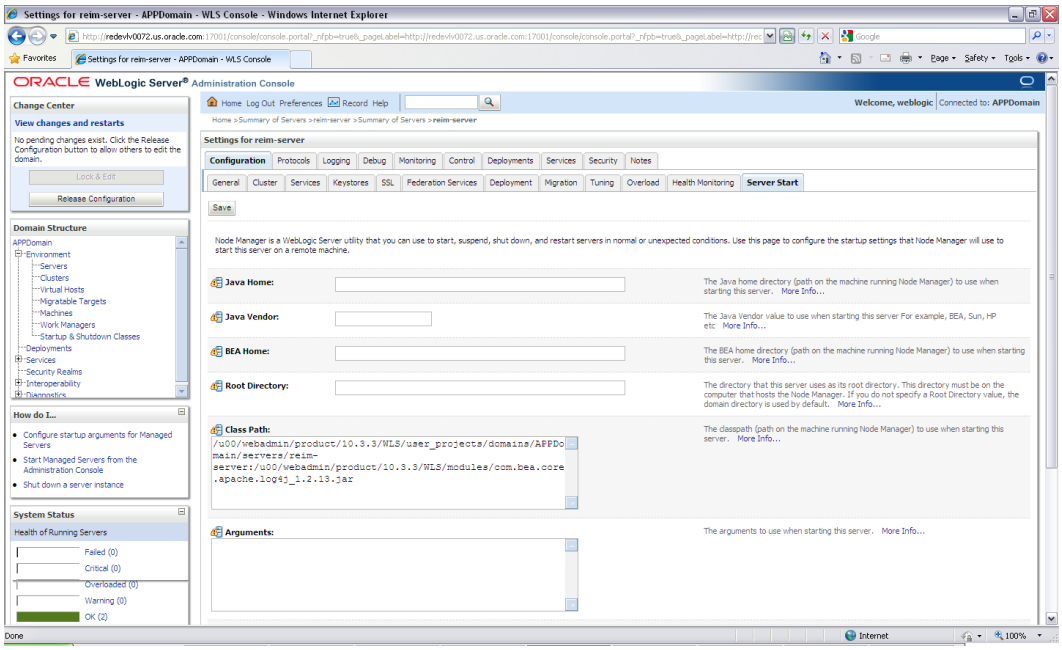
   After the Node Manager is started, the managed servers can be started through the Administration Console.
2. Navigate to Environments->Servers->select <app>-server server and click the Control tab.



3. Update in weblogic console->servers-><app>-server->server start tab->Classpath with the following:

   CLASSPATH:

   <WLS_HOME>/modules/com.bea.core.apache.log4j_1.2.13.jar



4. Export WEBLOGIC_DOMAIN_HOME=<WLS_HOME>/user_projects/domains/<domain name>
5. Update <WLS_HOME>/server/lib/weblogic.policy file with the below.

> **Note:** If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semicolon.

> **Note:** <WEBLOGIC_DOMAIN_HOME> in the below example is the full path of the Weblogic Domain, <managed_server> is the managed server created for the App and <context_root> correlates to the value entered for the application deployment name/context root of the application during installation. See the example. There should not be a space after **file:** in the following. file:<WEBLOGIC_DOMAIN_HOME>.

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

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

```
grant codeBase
"file:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/APPDomain/servers
/reim-server/tmp/_WL_user/reim01/-" {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";}
```

Expand the ReIM Application Distribution

To expand the ReIM application distribution, do the following.

1. Log in to the UNIX server as the user who owns the WebLogic installation. Create a new staging directory for the ReIM application distribution (reim13application.zip). There should be a minimum of 50 MB disk space available for the application installation files.

   Example: /u00/webadmin/media/reim

   This location is referred to as INSTALL_DIR for the remainder of this chapter.
2. Copy reim13application.zip to INSTALL_DIR and extract its contents.

Clustered Installations– Preinstallation Steps

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

1. Make sure that you are able to start and stop the managed servers that are part of the ReIM Cluster from the Administration Console.
2. Insert into $WEBLOGIC_HOME/wlserver_10.3/server/lib/weblogic.policy file, the same ReIM entries for java security permissions you entered on the main server. See the Start the Managed Servers section for additional information.

There are no additional steps to take before running the installer for ReIM.

Run the ReIM Application Installer

Once you have a managed server that is configured and started, you can run the ReIM application installer. This installer configures and deploys the ReIM application.

Note: See Appendix: ReIM Application Installer Screens for details on every screen and field in the application installer.

Note: It is recommended that the installer be run as the same UNIX account which owns the application server ORACLE_HOME files. This method takes full advantage of the installer's capabilities. If the installer is run as a different user, the Manual Deployment Option must be selected. However, note the following issue that describes consequences of Manual Deployment.

1. Change directories to INSTALL_DIR/reim/application.
2. Set the ORACLE_HOME and JAVA_HOME environment variables. ORACLE_HOME should point to your WebLogic 11g installation. JAVA_HOME should point to the Java 6.0 (1.6.0) JDK.
3. If you are using an X server such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
4. Run the install.sh script. This launches the installer. After installation is completed, a detailed installation log file is created (reim13install.<timestamp>.log).

   Note: The manual installation option in the installer is not functional for this release. See the section, "Files not available to copy at the end of installation results in non working applications" in Appendix E: Common Installation Errors.

5. Prior to running the ReIM batch programs, run the following command.

   ```
   cp
   $WEBLOGIC_DOMAIN_HOME/retail/reim13/properties/com/retek/reim/reim.properties
   $WEBLOGIC_DOMAIN_HOME/retail/reim13/batch/WEB-INF/classes/com/retek/reim/
   ```

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 a list of common installation errors.

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

Oracle Configuration Manager

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

The following document is available through My Oracle Support Access My Oracle Support at the following URL:

`https://support.oracle.com`

Oracle Configuration Manager Installer Guide (ID 1071030.1)

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

OCM Documentation Link

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

Clustered Installations– Post-Installation Steps

If you are installing the ReIM 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 node with the ORACLE_HOME you used for the ReIM installer is referred to as the *master server*. All other nodes are referred to as the *remote server*.

1. The ReIM batch files should be copied from the master server to each of the remote servers under the same path as on the master server. You should take the $WEBLOGIC_DOMAIN_HOME/retail/<reim application name>/batch directory and copy it onto the remote servers under the same path.
2. The Oracle Retail Installation creates some security files on $WEBLOGIC_DOMAIN_HOME/retail/<reim_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.

Backups Created by Installer

The ReIM application installer backs up a previous batch script installation by renaming it from reim-batch to reim-batch.<timestamp>. 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.

Example: reim-batch.200803011726

Test the ReIM Application

After the application installer completes you should have a working ReIM application installation. To launch the application, open a web browser and go to http://host:httpport/contextroot/index.jsp.

Example: http:// redevlv0072: 17009/reim01/index.jsp

Oracle Retail provides test cases that allow you to smoke test your installation. See the My Oracle support document, "Oracle Retail Merchandising Installation Test Cases" (ID 1277131.1).

reim.properties

The reim.properties file contains most of the settings for the ReIM application. Many properties in this file are set by the installer to get a working application up and running, but you may want to modify other settings in this file.

To modify settings in the properties file, you must redeploy the ReIM application. The properties values are stored in the templates/reim.properties file, which is in the directory where you expanded the ReIM installer files (for example, <INSTALL_DIR>/reim/application/templates/reim/properties, where <INSTALL_DIR> is the directory the application installer was unzipped).
Edit the reim.properties file to set the properties to the desired values. Then rerun the installer to deploy ReIM.

ReIM Batch Scripts

The ReIM application installer configures and installs the batch scripts under $ORACLE_HOME/user_projects/domains/<domain>/reim-batch.

The batch scripts are copies of the same generic file. Their file names determine which functionality is run. To run batch scripts, use the alias name provided in the installer when ReIM is installed, the one that is written out to the Java wallet (for example, reim_batchpgmname ADMIN).

For the scripts to run correctly, values for the following variables must be provided:

- ORACLE_HOME: WebLogic Home directory where the ReIM application has been deployed.
- JAVA_HOME: Java 6.0 (JDK 1.6.0) installation that typically is being used by the WebLogic Application Server.

> **Example:** export ORACLE_HOME=/u00/webadmin/product/10.3.3/WLS
> export JAVA_HOME=/u00/webadmin/product/10.3.3/jdk16
> export PATH=JAVA_HOME/bin:$PATH

Online Help

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

Single Sign-On

Skip this section if ReIM is not used within an Oracle Single Sign-On environment.

> **Note:** This section assumes the Oracle WebLogic Server has already been registered with the Oracle Single Sign-On server through the regsso.sh script. See Oracle Single Sign-On documentation for details.

To set up single sign-on, complete the following steps.

1. If you are using Oracle Retail Invoice Matching in an Oracle Single Sign-On environment, then the Invoice Matching root context must be protected. Modify the following files
   - mod_wl_ohs.conf located in <WEBLOGIC_HOME>/Oracle_WT1/instances/instance1/config/OHS/ohs1

```
LoadModule weblogic_module
"<WEBLOGIC_HOME>/Oracle_WT1/ohs/modules/mod_wl_ohs.so"
<IfModule weblogic_module>
      WebLogicHost host name
      WebLogicPort  admin port number
      MatchExpression *.jsp
</IfModule>
<Location /reim_sso >
     SetHandler weblogic-handler
</Location>
```

- mod_osso.conf located in <WEBLOGIC_HOME>/Oracle_WT1/instances/instance1/config/OHS/ohs1/moduleconf

```
LoadModule osso_module
"<WEBLOGIC_HOME>/Oracle_WT1/ohs/modules/mod_osso.so"
<IfModule mod_osso.c>
    OssoIpCheck off
    OssoIdleTimeout off
    OssoSecureCookies off
    OssoConfigFile
<WEBLOGIC_HOME>/Oracle_WT1/instances/instance1/config/OHS/ohs1/osso/osso.c
onf
<Location /reim_sso >
      WebLogicHost host name
      WebLogicPort port number of managed server
      require valid-user
      AuthType Osso
   </Location>
</IfModule>
```

2. The descriptor files web.xml and weblogic.xml need to adjusted to include entries related to security constraints. Your web.xml should look like this:

```
<security-constraint>
           <display-name>Security Constraint</display-name>
           <web-resource-collection>
              <web-resource-name>SecurePages</web-resource-name>
              <description>These pages are only accessible by authorized
users.</description>
              <url-pattern>/reim</url-pattern>
              <url-pattern>/reim/*</url-pattern>
              <http-method>GET</http-method>
           </web-resource-collection>
           <auth-constraint>
              <description>These are the roles who have access to
ReIM.</description>
              <role-name>Users</role-name>
           </auth-constraint>
           <user-data-constraint>
              <description>This is how the user data must be
transmitted.</description>
              <transport-guarantee>NONE</transport-guarantee>
           </user-data-constraint>
        </security-constraint>
        <login-config>
           <auth-method>CLIENT-CERT</auth-method>
           <realm-name>myrealm</realm-name>
        </login-config>
        <security-role>
           <description>These are the roles who have access to
ReIM.</description>
           <role-name>Users</role-name>
        </security-role>


Your weblogic.xml file should look like this:

<security-role-assignment>
      <role-name>Users</role-name>
      <principal-name>users</principal-name>
    </security-role-assignment>
    <security-permission>
```

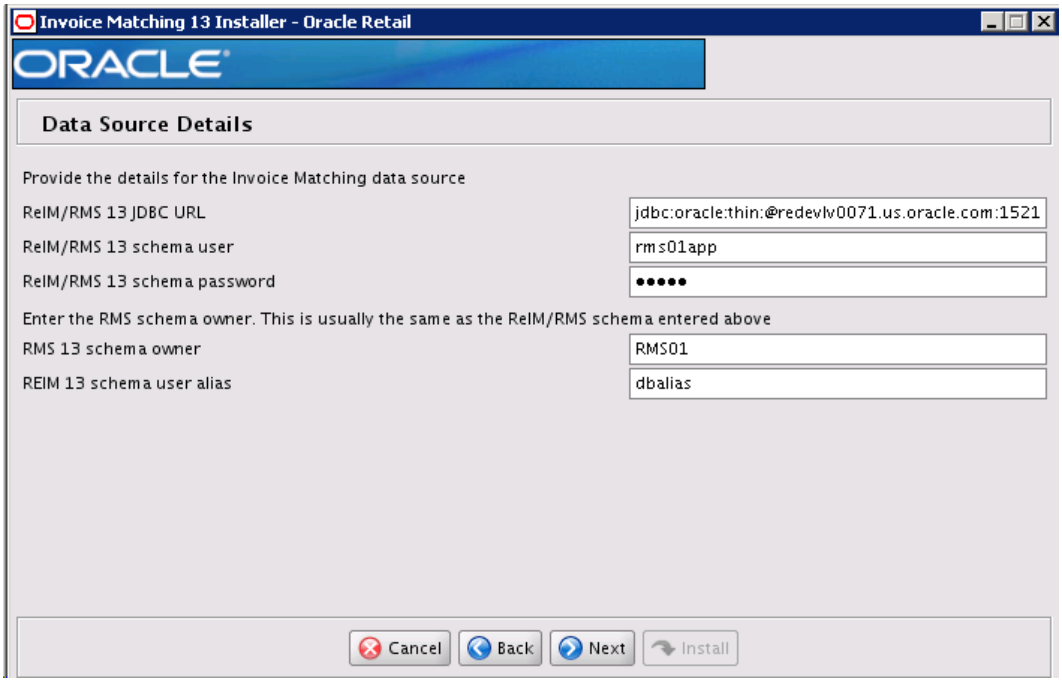
```
    <description>ReIM Security Permissions</description>
    <security-permission-spec>
        grant { permission java.net.SocketPermission "*", "resolve";  };
    </security-permission-spec>
</security-permission>
```

A

Appendix: ReIM Application Installer Screens

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

Screen: Data Source Details



Field Title	ReIM/RMS 13 JDBC URL
Field Description	URL used by the ReIM application to access the ReIM/RMS database schema. See Appendix: URL Reference for expected syntax.
Destination	reim.properties
Examples	jdbc:oracle:thin:@redevlv0071.us.oracle.com:1521:csols13

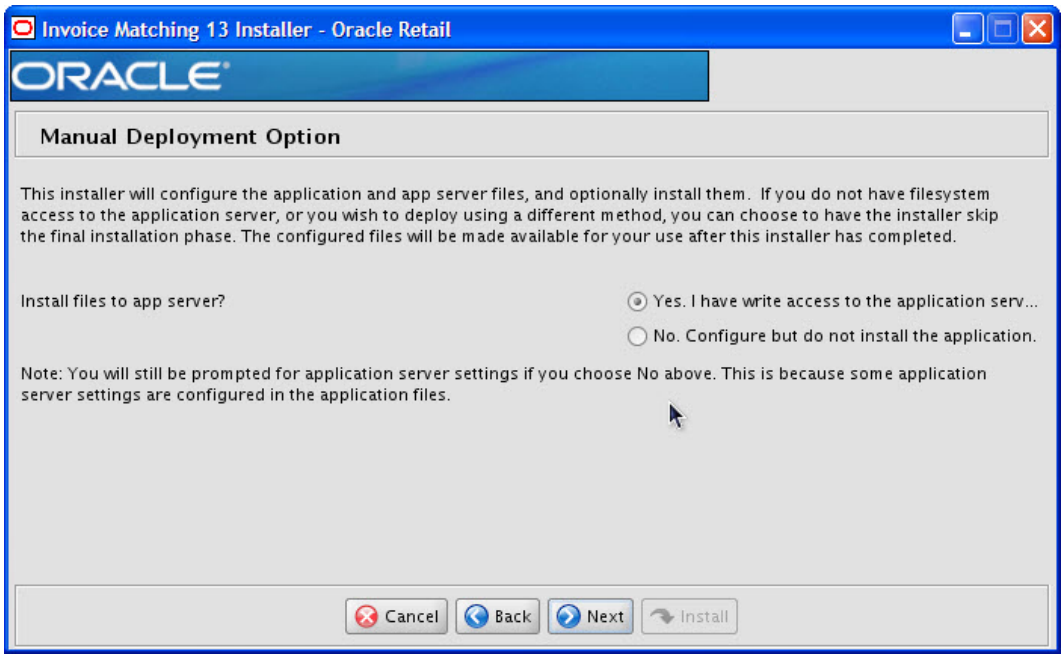
Field Title	ReIM/RMS 13 schema user
Field Description	RMS database user for accessing the ReIM tables. This should match what was given in the **RMS 13 schema** field of the ReIM database installer.
Destination	reim.properties
Example	rms01app

Field Title	ReIM/RMS 13 schema password
Field Description	Password for the JDBC username. This should match what was given in the **ReIM 13 schema password** field of the ReIM database installer.
Destination	wallet

Field Title	RMS 13 schema owner
Field Description	Database user which owns the RMS and ReIM tables. This usually has the same value as the **ReIM/RMS 13 schema** field above.
Destination	reim.properties
Example	RMS01

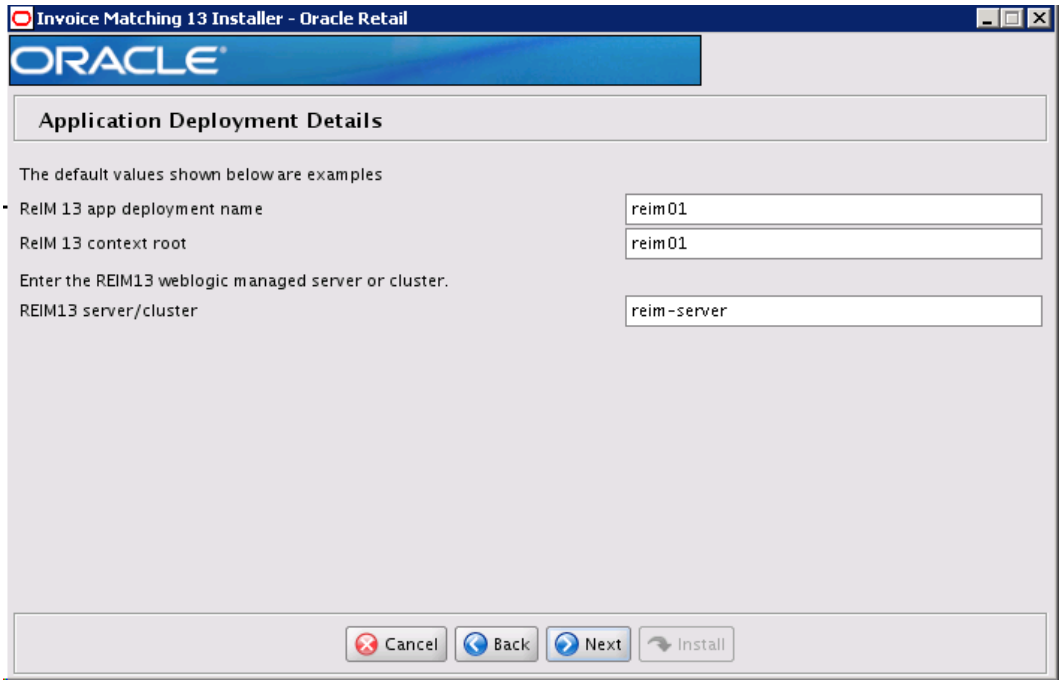
Field Title	REIM 13 schema user alias
Field Description	The alias of the ReIM user.
Destination	reim.properties
Example	db-alias

Screen: Manual Deployment Option



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

Screen: Application Deployment Details

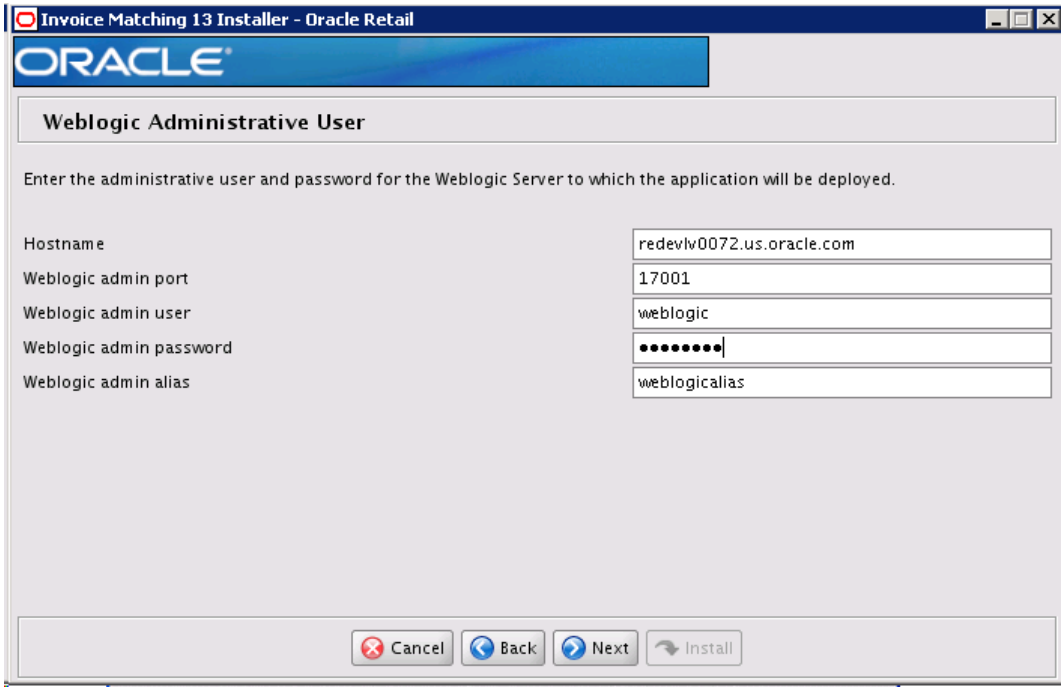


Field Title	ReIM 13 app deployment name
Field Description	Name by which this ReIM application is identified in the application server. This value must match the <context_root> added to the weblogic.policy file when the managed server for ReIM was created.
Example	reim01

Field Title	ReIM 13 context root
Field Description	Path under the HTTP URL used to access the ReIM application (for example, a context root of reim results in the application accessed at http://host:port/reim01/index.jsp). This value must match the <context_root> added to the weblogic.policy file when the managed server for ReIM was created.
Example	reim01

Field Title	ReIM 13 server/cluster
Field Description	Name of the ReIM WebLogic managed server or cluster.
Example	reim-server

Screen: WebLogic Administrative User



Field Title	Hostname
Field Description	Hostname of the application server
Example	redevlv0072.us.oracle.com

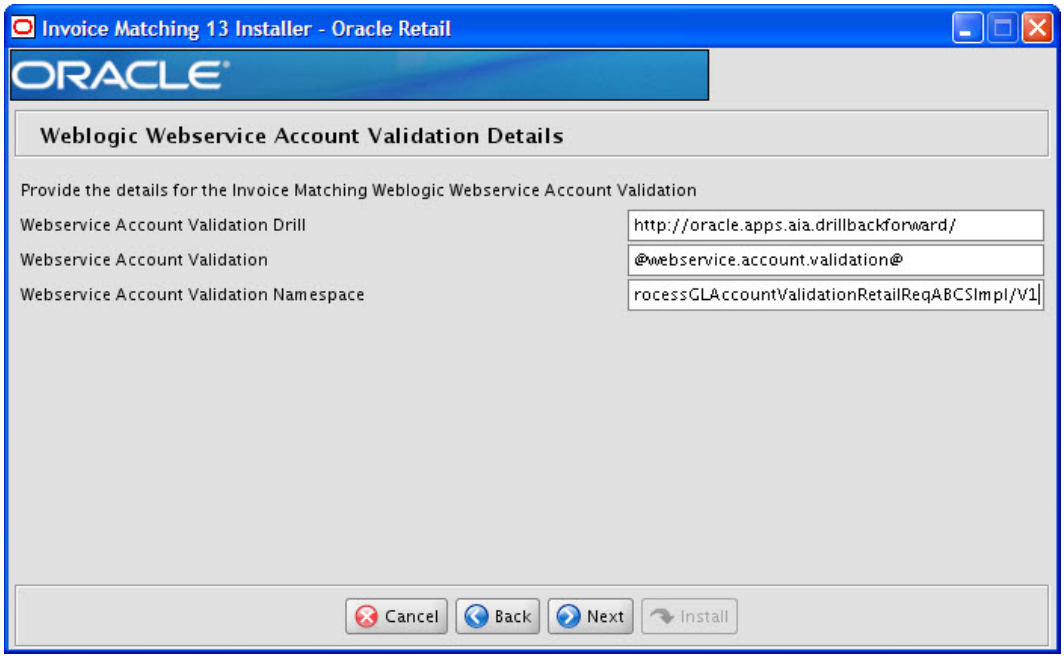
Field Title	WebLogic admin port
Field Description	This is the port of Administration Console.
Example	17001

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

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

Field Title	WebLogic admin alias
Field Description	An alias for the WebLogic admin user.
Example	weblogicalias

Screen: WebLogic Webservice Account Validation Details

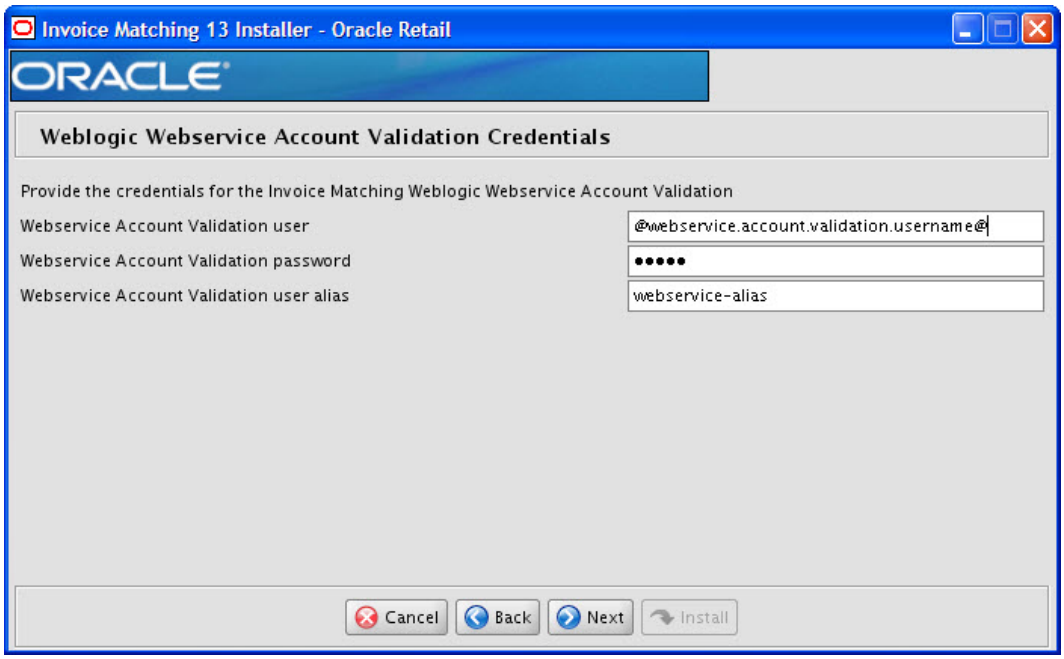


Field Title	Webservice Account Validation Drill
Field Description	The Web service provider URL used for drilling forward from the ReIM application. This information is from the financial application to which you are integrating (for example, PeopleSoft and Oracle E-Business Suite). Leave this field blank if there is no integration with a financial application.
Example	http://oracle.apps.aia.drillbackforward/

Field Title	Webservice Account Validation
Field Description	The URL for validating Web service accounts. This information is from the financial application to which you are integrating (for example, PeopleSoft and Oracle E-Business Suite). Leave this field blank if there is no integration with a financial application.
Example	@webservice.account.validation@

Field Title	Webservice Account Validation Namespace
Field Description	The URL for validating the Web service namespace. This information is from the financial application to which you are integrating (for example, PeopleSoft and Oracle E-Business Suite). Leave this field blank if there is no integration with a financial application.
Example	http://xmlns.oracle.com/ABCSImpl/Retail/Core/ProcessGLAccountValidationRetailReqABCSImpl/V1

Screen: WebLogic Webservice Account Validation Credentials

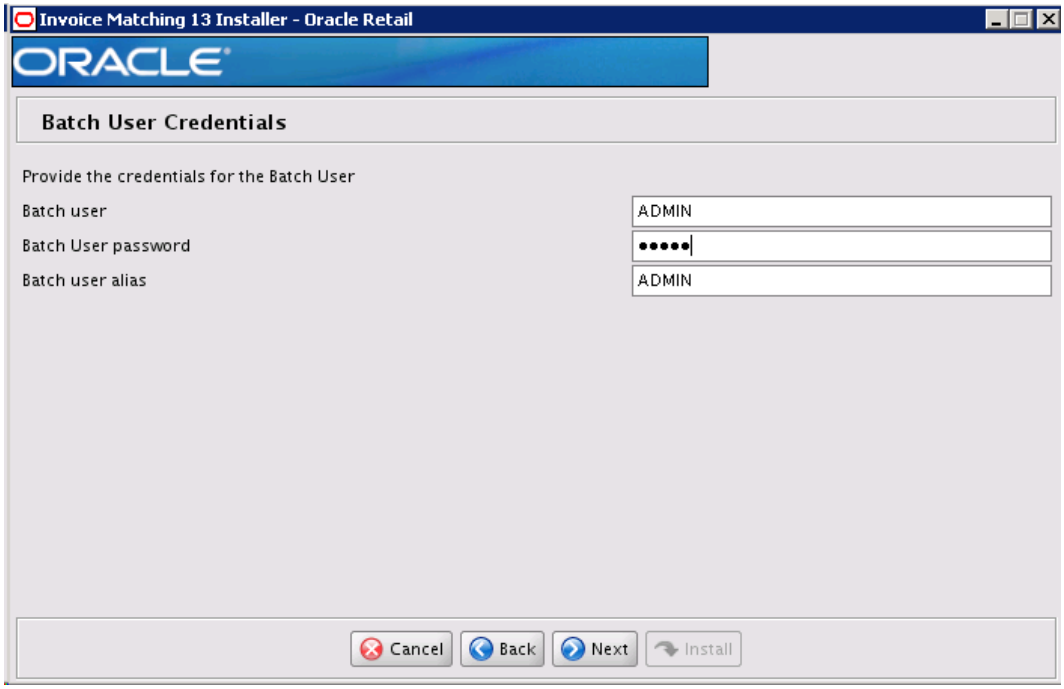


Field Title	Webservice Account Validation user
Field Description	The user for validating the Web service user name. A value is required in this field, even if you are not using Web service integration. The field is not validated, so enter any value.
Example	@webservice.account.validation.username@

Field Title	Webservice Account Validation Password
Field Description	The password for validating Web service accounts. A value is required in this field, even if you are not using Web service integration. The field is not validated, so enter any value.
Example	@webservice.account.validation.password@

Field Title	Webservice Account Validation Alias
Field Description	The alias for the Web service account user names A value is required in this field, even if you are not using Web service integration. The field is not validated, so enter any value.
Example	webservice-alias

Screen: Batch User Credentials



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

Field Title	Batch User Password
Field Description	The wallet password must match the database password on the database IM_USER_AUTHORIZATION table. The ReIM default scripts include User= ADMIN with and password=retek..

Field Title	Batch User Alias
Field Description	The alias for the user running ReIM batch. This alias is part of ORACLE wallet implementation. You will use this alias when running ReIM batch scripts.
Example	ADMIN

B

Appendix: Oracle Single Sign-On

Single Sign-On (SSO) is a term for the ability to sign onto multiple Web applications through a single user ID/Password. There are many implementations of SSO – Oracle currently provides three different implementations: Oracle Single Sign-On (OSSO), Java SSO (with the 10.1.3.1 release of OC4J) and Oracle Access Manager (provides more comprehensive user access capabilities).

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

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

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

What Do I Need for Oracle Single Sign-On?

The nexus of an Oracle Single Sign-On system is the Oracle Identity Management Infrastructure installation. This consists of the following components:

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle HTTP Server 11g Release 1 as a front end to the Oracle WebLogic Server. The Oracle HTTP Server is included in the Oracle Web Tier Utilities 11g Release 1 (11.1.1.3).
- An Oracle Single Sign-On Plug-in, used to authenticate the user and create the OSSO session cookie. This is available in the Oracle Fusion Middleware 11g Web Tier Utilities (11.1.1.30) package.
- The Delegated Administration Services (DAS) application, used to administer users and group information. This information may also be loaded or modified through standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OSSO system and registering HTTP servers.

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

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

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

Oracle Single Sign-on Terms and Definitions

This section provides definitions for terms pertaining to Oracle Single Sign-on.

Authentication

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

Dynamically Protected URLs

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

Identity Management Infrastructure

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

MOD_OSSO

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

MOD_WEBLOGIC

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

Oracle Internet Directory

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

Partner Application

A partner application is an application that delegates authentication to the Oracle Identity Management Infrastructure. One such partner application is the Oracle HTTP Server (OHS) supplied with the Oracle Application Server. OHS uses the MOD_OSSO module to configure this functionality.

All partner applications must be registered with the Oracle Single Sign-On server. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Realm

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

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

Statically Protected URLs

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

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

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

What Single Sign-On is not

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

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

How Oracle Single Sign-On Works

Oracle Single Sign-On involves several components. These are:

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

Statically Protected URLs

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

1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS).
2. The Oracle HTTP Server processses the request and routes it to the mod_oss module.

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

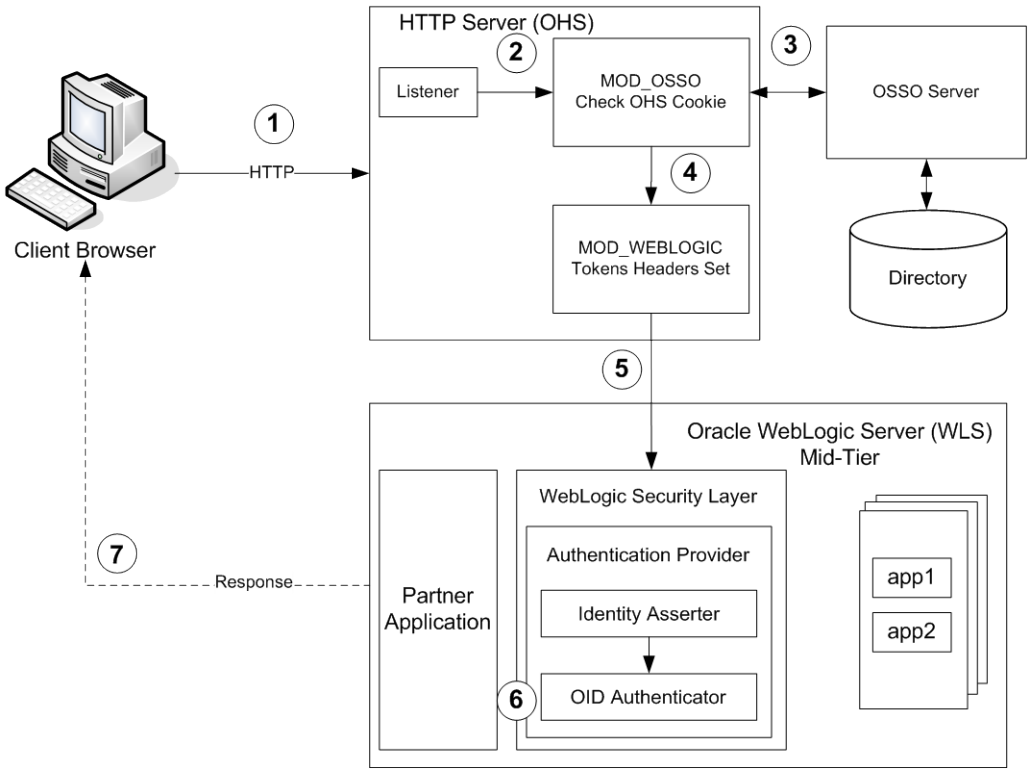
Dynamically Protected URLs

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

1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS). The Oracle HTTP server recognizes the user has not been authenticated, but allows the user to access the URL.
2. The application determines the user must be authenticated and send the Oracle HTTP Server a specific status to begin the authentication process.
3. The Oracle HTTP Server processses the request and routes it to the mod_oss module.
4. This module determines whether the user is already authenticated. If the authentication is required, it directs the browser to the OSSO server. The OSSO server checks for a secure cookie containing the authentication information. If the cookie is not found, the following occurs:
   a. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
   b. The user must sign in through a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server. The browser is then redirected back to the Oracle HTTP Server with the encrypted authentication credentials. It does NOT contain the user's password.
5. The mod_osso module then decrypts the user credentials and sets HTTP headers with relevant user attributes, marking the user's session as authenticated.

6. The mod_weblogic module (within the Oracle HTTP Server) then forwards the request to the Oracle WebLogic Server.
7. The Oracle WebLogic Server then invokes the configured authentication providers that decode the headers and provide the user's role membership. In an OSSO implementation, ensure that the OSSO Identity Asserter is invoked and Oracle Internet Directory (OID) Authenticator is executed to provide the user's role membership.
8. Once the authentication is established, the relevant application logic is initiated and the response is sent back to the user through the Oracle HTTP Server.
   Because the Web browser session is now authenticated, subsequent requests in that session are not redirected to the OSSO server for authentication.

Single Sign-on Topology



Installation Overview

Installing Oracle Single Sign-On consists of installing the following components:

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

   The Infrastructure OAS includes the Delegated Administration Services (DAS) application as well as the OSSO servlet. The DAS application can be used for user and realm management within OID.

2. Installing additional midtier instances (such as OAS 10.1.4) for the Oracle Retail applications, such as RMS, that are based on Oracle Forms technologies. These instances must be registered with the Infrastructure OAS installed in step 1.
3. Installing additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities.

Infrastructure Installation and Configuration

The Infrastructure installation for OSSO is dependent on the environment and requirements for its use. Deploying an Infrastructure OAS to be used in a test environment does not have the same availability requirements as for a production environment. Similarly, the Oracle Internet Directory (OID) LDAP server can be deployed in a variety of different configurations. See the *Oracle Application Server Installation Guide* and the *Oracle Internet Directory Installation Guide* for more details.

OID User Data

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

The current version of Oracle Single Sign-On only supports OID as its user storage facility. Customers with existing corporate LDAP implementations may need to synchronize user information between their existing LDAP directory servers and OID. OID supports standard LDIF file formats and provides a JNDI compliant set of Java classes as well. Moreover, OID provides additional synchronization and replication facilities to integrate with other corporate LDAP implementations.

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

OID with Multiple Realms

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

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

User Management

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

OID DAS

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

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

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

LDIF Scripts

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

User Data Synchronization

The user store for Oracle Single Sign-On resides within the Oracle Internet Directory (OID) LDAP server. Oracle Retail applications may require additional information attached to a user name for application-specific purposes and may be stored in an application-specific database. Currently, there are no Oracle Retail tools for synchronizing changes in OID stored information with application-specific user stores. Implementers should plan appropriate time and resources for this process. Oracle Retail strongly suggests that you configure any Oracle Retail application using an LDAP for its user store to point to the same OID server used with Oracle Single Sign-On.

C

Appendix: Installer Silent Mode

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

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

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

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

D

Appendix: URL Reference

Both the database schema and application installers for the Invoice Matching product require certain URLs, including 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

E

Appendix: Common Installation Errors

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

Database installer hangs on startup

Symptom

When the database schema installer is run, the following is written to the console and the installer hangs indefinitely:

```
Running pre-install checks
Running tnsping to get listener port
```

Solution

The installer startup script is waiting for control to return from the **tnsping** command, but tnsping is hanging. Type Control+C to cancel the installer, and investigate and solve the problem that is causing the tnsping <sid> command to hang. This can be caused by duplicate database listeners running.

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 than is supported by the installer. Set JAVA_HOME to the appropriate Java 1.6 (JDK 1.6.0) that is being used by the WebLogic Application Server and run the installer again.

Warning: Could not create system preferences directory

Symptom

The following text appears in the installer Errors tab:

```
May 22, 2006 11:16:39 AM java.util.prefs.FileSystemPreferences$3 run
WARNING: Could not create system preferences directory. System preferences are
unusable.
May 22, 2006 11:17:09 AM java.util.prefs.FileSystemPreferences
checkLockFile0ErrorCode
WARNING: Could not lock System prefs. Unix error code -264946424.
```

Solution

This is related to Java bug 4838770. The /etc/.java/.systemPrefs directory may not have been created on your system.

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

ConcurrentModificationException in Installer GUI

Symptom

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

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

Solution

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

Warning: Could not find X Input Context

Symptom

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

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

Solution

This message is harmless and can be ignored.

Warning: Lower case user IDS supplied with the application do not work

Symptom

The default user supplied with the ReIM application (for example, retail.user, where password = retek) does not work to for signing on to the application.

Solution

The user/password combination does not work because the password hashing is incorrect in the database scripts run by the installer code that affect only lower case user IDs/user names.

Run this code instead of the ReIM database to fix the passwords for the lower case user IDs supplied as part of the ReIM application:

update im_user_authorization iua

set password = reim_security_sql .hash(username, password) where lower (username) = username;

commit;

Files not available to copy at the end of installation results in non working applications

If you choose the option, **No. Configure but do not install the application** in the installer screen titled **Manual Deployment Option**, files that are required for manual application installation are deleted at the end of the installation.

Solution

Manual Deployment is not currently available in this installer.

Choose **Yes. I have write access to the application server** in the installer screen, **Manual Deployment Option**.

> **Note:** To successfully perform this option, you also need to run the installer as a user with write access to the WebLogic installation.

Installer fails because of missing .jar in $ORACLE_HOME/utils/ccr/lib

Symptom

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

Solution

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

GUI screens fail to open when running Installer

Symptom

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

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

Solution

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

F

Appendix: Setting Up Password Stores with Oracle Wallet

As part of an application installation, administrators must setup password stores for database user accounts using Oracle Wallet. These password stores must be installed on the application database side. While the installer handles much of this process, the administrators must perform some additional steps.

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

About Password Stores and Oracle Wallet

Oracle databases have allowed other users on the server to see passwords in case database connect strings (user name/password@db) were passed to programs. In the past, users could navigate to `ps -ef|grep <username>` and 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`, then the argument to a program would be the following:

```
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 need to use embedded usernames and passwords. This reduces any security risks that may exist because usernames and passwords are no longer exposed.

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

There are two different types of password stores or wallets. One type is for database connect strings used in program arguments (such as `sqlplus /@db_username`). The other type is for java application installation and application use.

Setting Up Password Stores for Database User Accounts

After the database is installed and the default database user accounts are set up, administrators must set up a password store using the Oracle Wallet. This involves assigning an alias for the username and associated password for each database user account. The alias is used later during the application installation. This password store must be created on the system where the application server and database client are installed.

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

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

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

1. Create a wallet using the following command:

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

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

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

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

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

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

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

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

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

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

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

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

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

Setting Up Wallets for Database User Accounts

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

- For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI
- For Java Applications: SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI

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

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

```
cd /projects/rms13.2/dev/
mkdir .wallet
```

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

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

```
WALLET_LOCATION =   (SOURCE =     (METHOD = FILE)     (METHOD_DATA =
(DIRECTORY =  /projects/rms13.2/dev/.wallet)) )
SQLNET.WALLET_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```

> **Note**: WALLET_LOCATION must be totally 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 = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))

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

> **Note**: It is important to not just copy the tnsnames.ora file because it can quickly become out of date. The ifile clause (shown above) is key.

4. Create the wallet files. These are empty initially.
   a. Ensure you are in the intended location.
      ```
      $ pwd
      /projects/rms13.2/dev/.wallet
      ```
   b. Create the wallet files.
      ```
      $ mkstore -wrl . –create
      ```
   c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.
   d. Enter password again.

      Two wallet files are created from the above command:
      - ewallet.p12
      - cwallet.sso
5. Create the wallet entry that associates the username and password to the custom tns alias that was setup in the wallet's tnsnames.ora file.

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

   Example: `mkstore –wrl . –createCredential dvols29_rms01user rms01user passwd`
6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.

   ```
   $ export TNS_ADMIN=/projects/rms13.2/dev/.wallet /* This is very import to use
   wallet to point at the alternate tnsnames.ora created in this example */

   $ sqlplus /@dvols29_rms01user

   SQL*Plus: Release 11

   Connected to:
   Oracle Database 11g

   SQL> show user
   USER is “rms01user”
   ```

   Running batch programs or shell scripts are 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

  ```
  oracle.security.client.connect_string1
  oracle.security.client.user1
  oracle.security.client.password1
  ```

- View the details of a wallet entry

  ```
  mkstore –wrl . –viewEntry oracle.security.client.connect_string1
  ```

  Returns the value of the entry:

  ```
  dvols29_rms01user
  mkstore –wrl . –viewEntry oracle.security.client.user1
  ```

  Returns value of the entry:

  ```
  rms01user

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

  Returns value of the entry:

  ```
  passwd
  ```

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

For Java application, consider the following:

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.
- Passwords are not used to update entries in Java wallets. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application that has been installed, the wallet is located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/config Example: mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config
- Application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.
- Scripts are located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/retail-public-security-api/bin for administering wallet entries.

  Example:

  mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/retail-public-security-api/bin
- In this directory is a script to help you update each alias entry without having to remember the wallet details. For example, if you set the RPM database alias to rms01user, you will find a script called update-RMS01USER.sh.

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

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

Example:

mspdv351:[1033_WLS] /u00/webadmin/reim/application/retail-public-security-api/bin

update-<ALIAS>.sh

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

Usage:

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

Example:

```
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.4/WLS/user_projects/domains/java_domain/retail/rpm132test/retail-public-security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass in.
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.4/WLS/user_projects/domains/java_domain/retail/rpm132test/retail-public-security-api/bin>
```

dump_credentials.sh

dump_credentials.sh is used to retrieve information from wallet. For each entry found in the wallet, the wallet partition, the alias, and the username 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/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

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

Below are the credentials found in the wallet at the location:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

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

Application level key partition name:reim13
User Name Alias:WLS-ALIAS User Name:weblogic
User Name Alias:RETAIL-ALIAS User Name:retail.user
User Name Alias:LDAP-ALIAS User Name:RETAIL.USER
User Name Alias:RMS-ALIAS User Name:rms132mock
User Name Alias:REIMBAT-ALIAS User Name:reimbat
```

save_credential.sh

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

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

Example:

```
mspdv351:[1033_WLS]
/u00/webadmin/mock132_testing/rtil/rtil/application/retail-public-security-
api/bin> save_credential.sh -l wallet_test -a myalias -p mypartition -u myuser

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

Enter password:
Verify password:
```

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

Usage:

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

How does the Wallet relate to the Application?

The ORACLE Retail Java applications include the wallet alias information you create in an <app-name>.properties file. Below is a sample from the reim.properties file. Database information and the user are presented below. The property called datasource.credential.alis=RMS-ALIAS uses the ORACLE wallet with the argument of RMS-ALIAS that is at the csm.wallet.path with the csm.wallet.partition.name of reim13 to retrieve the password for application use.

Reim.properties code sample:

```
datasource.url=jdbc:oracle:thin:@mspdv349.us.oracle.com:1521:pkols07
datasource.schema.owner=rms132mock
datasource.credential.alias=RMS-ALIAS
# ===============================================================
# ossa related Configuration
#
# These settings are for ossa configuration to store credentials.
# ===============================================================

csm.wallet.path=/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa
_domain/retail/reim13/config
csm.wallet.partition.name=reim
```

How does the Wallet relate to Java batch program use?

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

Setting up RETL Wallets

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

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

To setup RETL wallets, perform the following steps:

1. Set the following environment variables:
   - `ORACLE_SID=retaildb`
   - `RFX_HOME=/u00/rfx/rfx-13.2.0`
   - `RFX_TMP=/u00/rfx/rfx-13.2.0/tmp`
   - `JAVA_HOME=/usr/jdk1.6.0_12.64bit`
   - `LD_LIBRARY_PATH=$ORACLE_HOME`
   - `PATH=$RFX_HOME/bin:$JAVA_HOME/bin:$PATH`
2. Change directory to `$RFX_HOME/bin`.
3. Run `setup-security-credential.sh`.
   - Enter 1 to add a new database credential.
   - Enter the dbuseralias. For example, `retl_java_rms01user`.
   - Enter the database username. For example, `rms01user`.
   - Enter the database password.

- Reenter the database password.
- Enter D to exit the setup script.

4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

   For example, to configure RETLforRPAS, modify the following entries in `$MMHOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env`

   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     - `export RETL_WALLET_ALIAS="retl_java_rms01user"`
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     - `export ORACLE_WALLET_ALIAS="dvols29_rms01user"`
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     - `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"`

5. To change a password later, run `setup-security-credential.sh`.
   - Enter 2 to update a database credential.
   - Select the credential to update.
   - Enter the database user to update or change.
   - Enter the password of the database user.
   - Re-enter the password.

Quick Guide for Retail Wallets

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

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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/config							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
			<installed app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
RSL app	JAVA	<RSL INSTALL DIR>/rsl-rms/security/config							Each alias must be unique
			rsl-rsm	<rsl weblogic user alias>	<rsl weblogic user name>	App use	Installer	weblogic-alias	
			rsl-rsm	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
SIM app	JAVA	<weblogic domain home>/retail/<deployed sim app name>/config							
			rpm	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rms	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			rib-sim	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application
RETL	DB	<RETL home>/.wallet	n/a	<target application user alias>	<target application db userid>	App use	Manual	<db>_<user>	User may vary depending on RETL flow's target application
RIB	JAVA	<RIBHOME DIR>/deployment-home/conf/security							<app> is one of aip, rfm, rms, rpm, sim, rwms, tafr
JMS			jms<1-5>	<jms user alias> for jms<1-5>	<jms user name> for jms<1-5>	Integration use	Installer	jms-alias	
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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
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	

G

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 only some of the applications, the order is still valid, less the applications not being installed.

> **Note:** The installation order is not meant to imply integration between products.

Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM), Oracle Retail Sales Audit (ReSA). Optional: Oracle Retail Fiscal Management (ORFM)

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

2. Oracle Retail Service Layer (RSL)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Invoice Matching (ReIM)
7. Oracle Retail Price Management (RPM)

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

8. Oracle Retail Allocation
9. Oracle Retail Central Office (ORCO)
10. Oracle Retail Returns Management (ORRM)
11. Oracle Retail Back Office (ORBO) or Back Office with Labels and Tags (ORLAT)
12. Oracle Retail Store Inventory Management (SIM)

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

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

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