Oracle® Retail Service Layer

Installation Guide
Release 13.2
E22179-03

June 2011

ORACLE

Oracle® Retail Service Layer Installation Guide, Release 13.2

Primary Author: 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 Service Layer Installation Guide, Release 13.2

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

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

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

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

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

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

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

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

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

Preface

The Oracle Retail Service Layer development team delivers RSL applications only for non-Java/Java EE service providers such as RMS (Oracle Forms based). Other implementations of RSL exist but are bundled within other Oracle Retail Java EE applications and are installed as part of those applications. Currently RSL provides an implementation for integrating applications with RMS. RPM is the only Oracle Retail Java EE application that includes an integrated implementation of RSL.

An application-specific version of RSL in a Java EE environment is referred to as RSLfor<App> (for this release only RSLforRMS). This application is packaged as an EAR file that must be deployed in an application server. Currently, RSL applications have been certified to install and execute in Oracle Fusion Middleware 11g Release 1 (11.1.1.3) /Oracle WebLogic Server 11g Release 1 (10.3.3).

Audience

This manual is designed for System Administrators, Developers, and Applications Support personnel installing the RSL for RMS implementation provided by the RSL team.

Related Documents

For more information, see the following documents in the Oracle Retail Service Layer Release 13.2 documentation set:

- *Oracle Retail Service Layer Release Notes*
- *Oracle Retail Service Layer Programmer's Guide*

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."

> **Note:** This is a note. It is used to call out information that is important, but not necessarily part of the procedure.

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

A hyperlink appears like this.

1

Preinstallation Tasks

This chapter includes tasks to complete before 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 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 3 (OL5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) 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)

Check Oracle Retail Software Dependencies

Service Providing Applications (such as RMS) must have all RSL components installed (including stored procedures, tables, and Oracle objects). Refer to the installation guide for each for details.

Supported Oracle Retail Products

Integrates with	Version
Oracle Retail Allocation (Client)	13.2
Oracle Retail Store Inventory Management (SIM)	13.2.1
Oracle Retail Merchandising System (RMS) (Server)	13.2
Oracle Retail Price Management (RPM) (Server)	13.2

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.

Moving the RSL Packaged .jars to the Server Library

Copy the ojdbc6.jar from this location (RSL1320forRMS/rsl-rms/oracle) to this location ($WLS_HOME/Middleware/wlserver_10.3/server/lib).

Configure the rsl-rms-wls-instance

To configure the rsl-rms-wls-instance, do the following.

1. Configure the startup script.
   a. Take a backup of the script, $DOMAIN_HOME/base_domain/bin/startWebLogic.sh.
   b. Edit the script, $DOMAIN_HOME/base_domain/bin/startWebLogic.sh, to add the following attributes.

```
CLASSPATH=$DOMAIN_HOME/servers/$SERVER_NAME:$CLASSPATH
JAVA_VM="-server"
USER_MEM_ARGS="-Xms1024m -Xmx2048m -XX:MaxPermSize=256m"
```

   The following is a sample from startWebLogic.sh:

```
echo "."

echo "."

echo "JAVA Memory arguments: ${MEM_ARGS}"

echo "."

echo "WLS Start Mode=${WLS_DISPLAY_MODE}"

echo "."

CLASSPATH=$DOMAIN_HOME/servers/$SERVER_NAME:$CLASSPATH
JAVA_VM="-server"
USER_MEM_ARGS="-Xms1024m -Xmx2048m -XX:MaxPermSize=256m"

echo "CLASSPATH=${CLASSPATH}"

echo "."

echo "PATH=${PATH}"

echo "."

echo "*************************************************"

echo "*  To start WebLogic Server, use a username and   *"

echo "*  password assigned to an admin-level user.  For *"

echo "*  server administration, use the WebLogic Server *"

echo "*  console at http://hostname:port/console        *"

echo "*************************************************"

# CLASS CACHING

if [ "${CLASS_CACHE}" = "true" ] ; then
    echo "Class caching enabled..."
    JAVA_OPTIONS="${JAVA_OPTIONS} -Dlaunch.main.class=${SERVER_CLASS} -
Dlaunch.class.path="${CLASSPATH}" -
Dlaunch.complete=weblogic.store.internal.LockManagerImpl -cp
${WL_HOME}/server/lib/pcl2.jar"
```

```
    export JAVA_OPTIONS
    SERVER_CLASS="com.oracle.classloader.launch.Launcher"
fi

# START WEBLOGIC

echo "starting weblogic with Java version:"

${JAVA_HOME}/bin/java ${JAVA_VM} -version

if [ "${WLS_REDIRECT_LOG}" = "" ] ; then
    echo "Starting WLS with line:"
    echo "${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} -
Dweblogic.Name=${SERVER_NAME} -
Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}"
    ${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -
Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}
else
    echo "Redirecting output from WLS window to ${WLS_REDIRECT_LOG}"
    ${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -
Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${PROXY_SETTINGS} ${SERVER_CLASS}  >"${WLS_REDIRECT_LOG}" 2>&1
fi

stopAll
```

> **Note:** The statements above must be added to the startWebLogic script before the call is made to start the server.

2. Make changes to the nodemanager.properties file

   Edit this file as indicated below:

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

   a. Change the StartScriptEnabled property to True.

   b. Set the StartScriptName property to startWebLogic.sh.

   The following is a sample from the file.

```
StartScriptName=startWebLogic.sh
StartScriptEnabled=true
```

> **Note:** The nodemanager.properties file is created after the nodemanager is started for the first time. It is not available before that.
>
> **Note:** The nodemanager must be restarted after changes are made to the nodemanager.properties file.

2

Application Installation

Before proceeding, you must install WebLogic Application. The RSLforRMS application is deployed to an instance within the WebLogic Application installation.

Create a Managed Instance for RSLfor<App>

The RSLforRMS application must be deployed to its own dedicated WebLogic instance.

Use the following steps to create a new managed server instance for RSLforRMS and configure it to RSL requirements.

> **Note:** For information on using commands to create a managed server instance, see the Weblogic® Application Server Administrator's Guide 11g Release 3 (10.3.3).

Create the rsl-rms-wls-instance using WebLogic admin console GUI:

1. Log in to the WebLogic admin console GUI (http://<host>:<port>/console) as administrator.
2. In the right menu, navigate to Environment → Servers.
3. Click **New**.
4. Enter the Name, Port, and Listen address of the server instance to be created. For example:
   - Server Name : rsl-rms-wls-instance
   - Server Listen Address: myhost82

     > **Note:** The RSL application server must be on the same application server of the application that will be using RSL (for example, SIM, RPM, and Allocation).

   - Server Listen Port: 19007
5. Click **Next**. Click **Finish**.
6. Make sure you see the new instance listed under Servers.
7. Click on the server name that you just created. Navigate to Configuration → General tab.
8. In the Machine field, select the machine name where the server will be running.

Expand the RSLforRMS Distribution

To expand the RSLforRMS distribution, do the following.

1. Log into the UNIX server as the user who owns the WebLogic installation and determine where the RSL 13.2 application server file (RslServerPak13.2.0forRMS_eng_ga.tar) will be installed. There should be a minimum of 25 MB disk space available for the application installation files.
2. Copy RslServerPak13.2.0forRMS_eng_ga.tar (located at CD/appserverunix) to a newly created staging directory on the UNIX server.

3. Change directory to the location of RslServerPak13.2.0forRMS_eng_ga.tar and extract this file. This will create directory: /RSL1320forRMS.

   To extact run the following: tar xf RslServerPak13.2.0forRMS_eng_ga.tar

   RSL_INSTALL_HOME refers to the directory structure including the newly created /RSL1320forRMS.

   For example, /u00/webadmin/RSL_INSTALL/RSL1320forRMS.

Run the RSLforRMS Installer

Configuration scripts are provided to deploy and configure the RSLforRMS application in the application server, including the JDBC DataSource.

1. Change directory to RSL_INSTALL_HOME.
2. Set the WL_HOME, WEBLOGIC_DOMAIN_HOME and JAVA_HOME environment variables.

   WL_HOME should point to your Application Server installation (for example, WL_HOME = /u01/rrtswls/Oracle/Middleware. JAVA_HOME should point to the Java 6.0 (1.6.0).

   WEBLOGIC_DOMAIN_HOME should point to the application server domain (for example, WEBLOGIC_DOMAIN_HOME = /u00/webadmin/product/10.3/WLS/user_projects/domains/base_domain).
3. Set the PATH environment variable (for example, PATH=$JAVA_HOME/bin:$PATH).
4. If you are using an X server, such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
5. Run the rsl-installer.sh script. (See Appendix: RSLfor<App> Installer Screens.) This launches the installer. After installation is complete, a detailed installation log file is created (rsl13install.<timestamp>.log) in the RSL_INSTALL_HOME/.retail-installer directory.
6. After the script has run successfully, verify that the application is running and the Oracle DataSource was configured properly by logging into the Application Server Console.

Resolving Errors Encountered During Application Installation

If the application installer encounters any errors, it halts execution immediately.

See Appendix: Common Installation Errors for common installation errors.

Because the application installation is a full reinstallation every time, any previous partial installation will be overwritten by the successful installation.

Backups Created by Installer

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

For example, rsl-rms.200605011726.

A

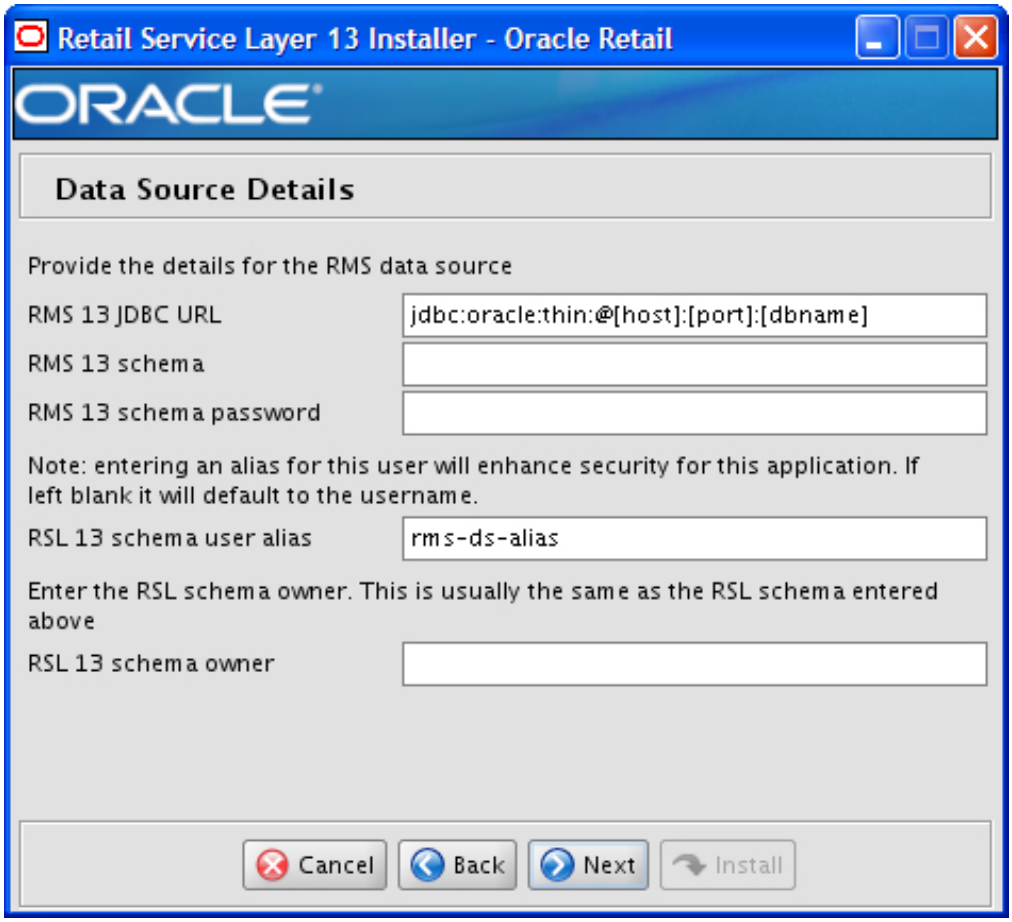
Appendix: RSLforRMS Installer Screens

This section describes the Oracle Retail Service Layer 13.2 Installer screens.

This environment information is necessary for successful deployment of the RSLforRMS application. The screens and fields displayed depend on the options selected; some screens and fields may not be displayed.

The following are RSLforRMS installer screens.

Screen: Data Source Details



Field Title	RMS 13 JDBC URL
Field Description	URL used by RSLforRMS to access the RMS database schema. See Appendix: URL Reference for expected syntax.
Example	jdbc:oracle:thin:@myhost:1525:mydatabase

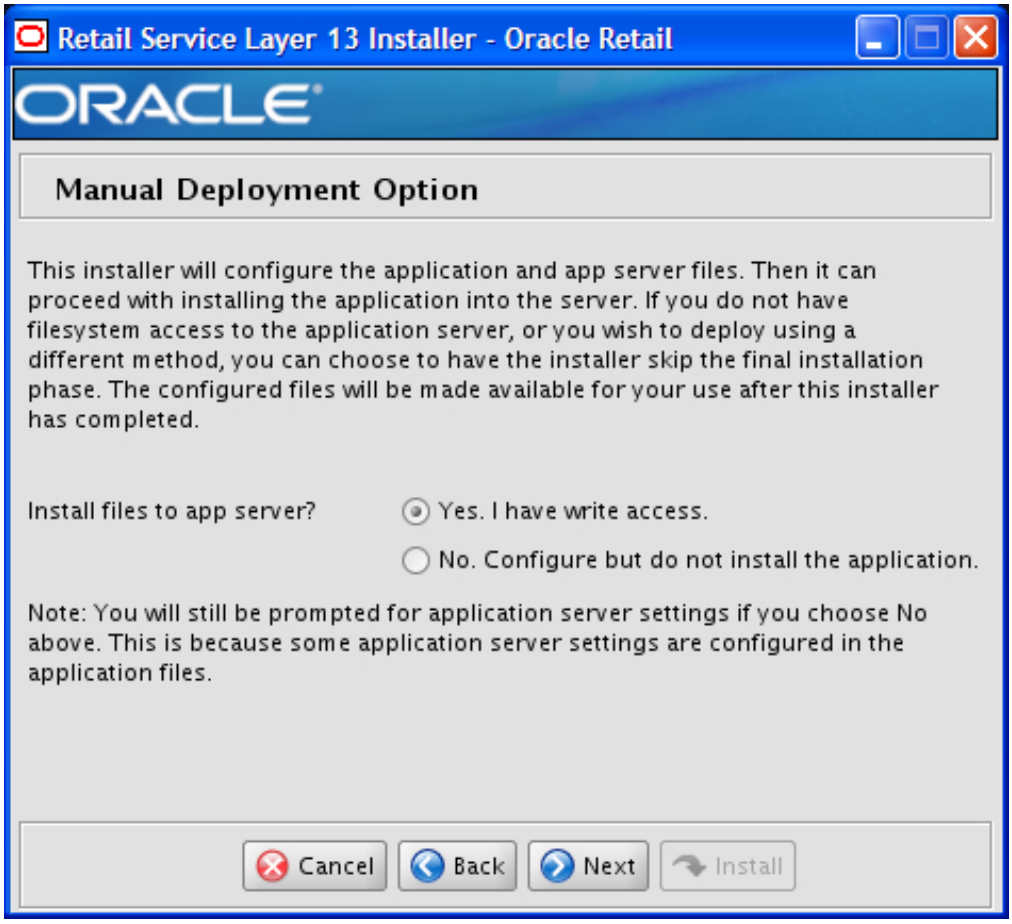
Field Title	RMS 13 schema
Field Description	Database schema owner user where the RMS database schema was installed.
Example	RMS13

Field Title	RMS 13 schema password
Field Description	Password for the RMS schema user.

Field Title	RSL 13 schema user alias
Field Description	Alias for the RMS/RSL schema user.

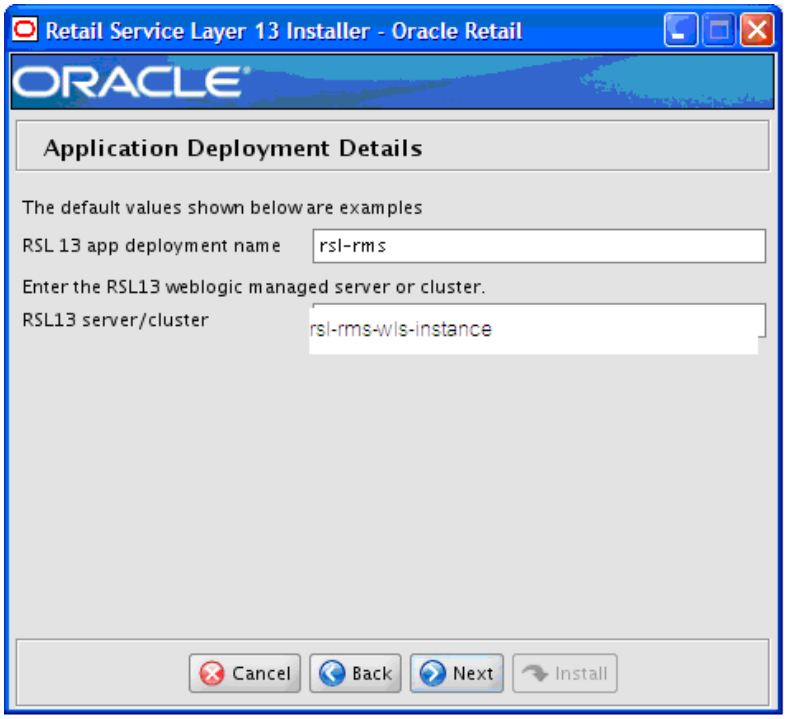
Field Title	RSL 13 schema owner
Field Description	Owner of the RMS schema.
Example	RMS13

Screen: Manual Deployment Option



Field Title	Install files to app server?
Field Description	If you do not have write access under WL_HOME, you can still use the installer to gather your settings and configure the RSLforRMS files locally in the staging area. Then, at a later time, an administrator can manually copy over the RSLforRMS files and deploy the ear file. If you select this option, instructions are printed to the console and the installer log file for the steps needed to complete the installation.
Example	Yes
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**, wallet files required for application run time are deleted at the end of the installation.

Screen: Application Server Instance Details



Field Title	RSL 13 app deployment name
Field Description	The name of the deployment/context_root.
Example	rsl-rms

Field Title	RSL 13 server/cluster
Field Description	The name of the RSL13 WebLogic managed server or cluster.
Example	rsl-rms-wls-instance

Screen: WebLogic Administrative User

Field Title	Hostname
Field Description	Host name of the WebLogic application server where RSL would be deployed.
Example	Myhost82 **Note:** The RSL application server must be on the same application server of the application that will be using RSL (for example, SIM, RPM, and Allocation.

Field Title	Weblogic admin port
Field Description	The port of the WebLogic admin server.
Example	7001

Field Title	Weblogic admin user
Field Description	The user name of the WebLogic server. The user should have administrator privileges.
Example	weblogic

Field Title	Weblogic admin password
Field Description	The password for the WebLogic user name.
Example	welcome1

Field Title	Weblogic admin alias
Field Description	The alias for the admin user name.
Example	Weblogic-admin-alias

B

Appendix: Installer Silent Mode

In addition to the GUI and text interfaces of the RSLforRMS 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, perform the following procedure:

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

   Example:

   ```
   rsl-installer.sh silent
   ```

C

Appendix: Common Installation Errors

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

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 $WL_HOME/jdk160_18 from the WebLogic Application Server 10.3.3 installation 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. See http://bugs.sun.com for details.

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

ConcurrentModificationException in Installer GUI

Symptom

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

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

Solution

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

Warning: Could not find X Input Context

Symptom

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

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

Solution

This message is harmless and can be ignored.

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**, wallet files that are required for application run time 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.

D

Appendix: URL Reference

The application installers for the RSLforRMS product will ask for several different URLs. These include the following.

JDBC URL for a Database

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

Syntax

`jdbc:oracle:thin:@<host>:<port>:<sid>`

where:

- `<host>` is the hostname of the database server.
- `<port>` is the database listener port.
- `<sid>` is the system identifier for the database.

Example:

`jdbc:oracle:thin:@myhost:1521:mysid`

E

Appendix: Setting Up Password Stores with Oracle Wallet

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

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

About Password Stores and Oracle Wallet

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

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

Users can retrieve the credentials using aliases that were set up when encrypting and storing the user credentials in the password store. For example, if `username/password@db` is entered in the command line argument and the alias is called `db_username`, 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 needed to use embedded usernames and passwords. This reduces any security risks that may exist because usernames and passwords are no longer exposed.

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

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

Setting Up Password Stores for Database User Accounts

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

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

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

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

1. Create a wallet using the following command:

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

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

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

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

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

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

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

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

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

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

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

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

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

Setting Up Wallets for Database User Accounts

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

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

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

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

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

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

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

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

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

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

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

```
ifile = /u00/oracle/product/11.2.0.1/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 the password again.

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

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

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

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

   $ sqlplus /@dvols29_rms01user

   SQL*Plus: Release 11

   Connected to:
   Oracle Database 11g

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

   Running batch programs or shell scripts is similar:

   ```
   Ex: dtesys /@dvols29_rms01user
   script.sh /@dvols29_rms01user

   Set the UP unix variable to help with some compiles :

   export UP=/@dvols29_rms01user
   for use in RMS batch compiles, and RMS, RWMS, and ARI forms compiles.
   ```

   As shown in the example above, users can ensure that passwords remain invisible.

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

  ```
  mkstore –wrl . –deleteCredential dvols29_rms01user
  ```
- Change the password for a credential on wallet

  ```
  mkstore –wrl . –modifyCredential dvols29_rms01user rms01user passwd
  ```

- List the wallet credential entries

  ```
  mkstore –wrl . –list
  ```

  This command returns values such as the following.

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

- View the details of a wallet entry

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

  Returns the value of the entry:

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

  Returns value of the entry:

  ```
  rms01user
  ```

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

  Returns value of the entry:

  ```
  passwd
  ```

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

For Java application, consider the following:

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.

  Java wallets do not have a password to update their entries. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application, after you unzip RMS13application.zip, cd into RMS/application/retail-public-security-api/bin (for example, mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/reim13/wallet/bin)

  or

  unzip to reim/application/retail-public-security-api-bin to run the commands below to adminster java wallets. The application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.

  There are two scripts relating to this at that folder: dump_credentials.sh and save_credential.sh.

dump_credentials.sh

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

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

Example:

```
dump_credentials.sh
location:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_do
main/sim-client/csm

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

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

Apapplication 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 tie back to the Application?

The ORACLE Retail Java applications have the wallet alias information you create in an <app-name>.properties file. Below is the reim.properties file. Note the database information and the user are presented as well. The property called datasource.credential.alias=RMS-ALIAS uses the ORACLE wallet with the argument of RMS-ALIAS at the csm.wallet.path and csm.wallet.partition.name = reim13 to retrieve the password for application use.

Reim.properties code sample:

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

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

How does the Wallet tie back to Java pgm batch use (such as ReIM batch)?

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 for 13.2

RETL 13.2 creates a wallet under $RFX_HOME/etc/security, with the following files:

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

To set up RETL wallets, perform the following steps:

1. Set the following environment variables:
   - `ORACLE_SID=retaildb`
   - `RFX_HOME=/u00/rfx/rfx-13.2.0`
   - `RFX_TMP=/u00/rfx/rfx-13.2.0/tmp`
   - `JAVA_HOME=/usr/jdk1.6.0_12.64bit`
   - `LD_LIBRARY_PATH=$ORACLE_HOME`
   - `PATH=$RFX_HOME/bin:$JAVA_HOME/bin:$PATH`
2. Change directory to `$RFX_HOME/bin`.
3. Run `setup-security-credential.sh`.
   - Enter 1 to add a new database credential.
   - Enter the dbuseralias. For example, `retl_java_rms01user`.
   - Enter the database user name. For example, `rms01user.`
   - Enter the database password.
   - Re-enter the database password.
   - Enter D to exit the setup script.
4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

   For example, to configure RETLforRPAS, modify the following entries in `$MMHOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env.`
   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     `export RETL_WALLET_ALIAS="retl_java_rms01user"`
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     `export ORACLE_WALLET_ALIAS="dvols29_rms01user"`
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"`
5. To change a password later, run `setup-security-credential.sh`.
   - Enter 2 to update a database credential.
   - Select the credential to update.
   - Enter the database user to update or change.
   - Enter the password of the database user.
   - Re-enter the password.

Quick Guide for Retail Wallets

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

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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbatch-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>/sim-client/csm							
			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							RMS is one of these: 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>	Integra-tion use	Installer	jms-alias	
Weblogic			rib-RMS-app-server-instance	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integra-tion use	Installer	weblogic-alias	
Admin GUI			rib-RMS#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integratio n use	Installer	admin-gui-alias	
Application			rib-RMS#user-alias	<app weblogic user alias>	<app weblogic user name>	Integratio n use	Installer	app-user-alias	Valid only for aip, rpm, sim
DB			rib-RMS#app-db-user-alias	<rib-app database schema user alias>	<rib-app database schema user name>	Integratio n 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-RMS#hosp-user-alias	<rib-app error hospital database schema user alias>	<rib-app error hospital database schema user name>	Integration use	Installer	hosp-user-alias	

F

Appendix: Installation Order

This section provides a guideline as to the order in which the Oracle Retail applications should be installed. If a retailer has chosen to use some, but not all, of the applications the order is still valid less the applications not being installed.

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

Enterprise Installation Order

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

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

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

   > **Note:** During installation of RPM, you are asked for the RIBforRPM provider URL. Since RIB is installed after RPM, make a note of the URL you enter. If you need to change the RIBforRPM provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

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. Since RIB is installed after SIM, make a note of the URL you enter. If you need to change the RIB provider URL after you install RIB, you can do so by editing 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 Workspace (ORW)