Oracle® Retail Active Retail Intelligence

Installation Guide
Release 14.1
E59140-02

March 2016

ORACLE

Oracle® Active Retail Intelligence Installation Guide, Release 14.1

Primary Author: Wade Schwarz, Mourya Pantham

Contributors: Nathan Young

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments

Oracle Retail Active Retail Intelligence Installation Guide, Release 14.1

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 Active Retail Intelligence Release 14.1 documentation set:

- *Oracle Retail Active Retail Intelligence Release Notes*
- *Oracle Retail Active Retail Intelligence User Guide*
- *Oracle Retail Active Retail Intelligence Online Help*
- *Oracle Retail Active Retail Intelligence Operations Guide*
- *Oracle Retail Merchandising Implementation Guide*
- *Oracle Retail Merchandising Security 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, 14.1) or a later patch release (for example, 14.1.1). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

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

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

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

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

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

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

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

Conventions

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

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

1

Preinstallation Tasks

Check Supported Database Server Requirements

General Requirements for a database server running ARI include:

Supported on:	Versions Supported:
Database Server OS	OS certified with Oracle Database 12cR1 Enterprise Edition. Options are: ▪ Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Database Server 12cR1	Oracle Database Enterprise Edition 12cR1 (12.1.0.1.4) with the following specifications: **Components:** ▪ Oracle Partitioning ▪ Examples CD **Patches:** ▪ 18522516: 12.1.0.1.4 Database Patch Set Update. ▪ 18705901: 12.1.0.1.4 Database Patch Set Update for Grid Infrastructure. **Oneoffs:** ▪ 18169693: ORA-28595: Extproc agent: Invalid DDL Path. ▪ 17815049: ORA-600 [KPONMARKCONN1] WHEN STARTING INSTANCE ▪ Patch 19623450: MISSING JAVA CLASSES AFTER UPGRADE TO JDK 7 ▪ 18404105: GETTING ORA-22345 WHILE TRYING TO RECOMPILE THE TYPE USING EXECUTE IMMEDIATE STM. **Other components:** ▪ Perl interpreter 5.0 or later ▪ X-Windows interface ▪ JDK 1.7

Note: By default, JDK is at 1.6. After applying the 12.1.0.1.4 patchset, follow the instructions on Oracle Database Java Developer's Guide 12c Release 1 to upgrade JDK to 1.7, then apply patch 19623450. The Guide is available here:

http://docs.oracle.com/database/121/JJDEV/chone.htm#JJDEV01000

Check Supported Application Server Requirements

General requirements for an application server capable of running ARI include:

Supported on	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.7). Options are: ▪ Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Application Server	Oracle Fusion Middleware 11g Release 2 (11.1.2.2) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.6) ▪ Oracle Forms Services 11g Release 2 (11.1.2.2) **Java:** ▪ JDK 1.7.0+ 64 bit **Optional (SSO required)** ▪ Oracle Access Management 11gR2 (11.1.2.2) ▪ Oracle Internet Directory 11.1.1.7 ▪ OHS 11.1.1.7 with WebGate Agent 11gR2(11.1.2.2) ▪ Must have separate WebLogic 10.3.6 for Oracle Access Manager 11gr2.

Verify Single Sign-On

If ARI is not being deployed in a Single Sign-On environment, skip this section.

If Single Sign-On is to be used, verify that Oracle Identity and Access Management 11gR2 version 11.1.2.2 has been installed along with the components listed in the above Application Server requirements section. Verify the HTTP Server is registered with the Oracle Access Manager (OAM) 11gR2 as a partner application.

Check Supported Web Browser and Client Requirements

General requirements for client running ARI include:

Requirement	Version
Operating system	Windows 7 or 8
Display resolution	1024x768 or higher
Processor	2.6GHz or higher
Memory	1GByte or higher
Networking	intranet with at least 10Mbps data rate
Oracle (Sun) Java Runtime Environment	1.8
Browser	Microsoft Internet Explorer 11 **Note:** If you are using Internet Explorer 11 with JRE 1.8, it is recommended that you install the 32 bit version of JRE 1.8 in the client system. ARI may not work when the 64 bit JRE version is installed on the client system." Mozilla Firefox 24.0

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)/Oracle Retail Trade Management (RTM)/Oracle Retail Sales Audit (ReSA)	14.1

Create a UNIX User Account to Install the Software

1. Create a UNIX group named "dev".
2. Create UNIX user named "oretail" and assign it to the "dev" group. This user will install the ARI software

Create Staging Directory for ARI Database Files

1. Create a staging directory for the ARI database installation software. There should be a minimum of 50 MB disk space available.
2. Copy the ari14dbserverunix.zip file from the CDROM directory to the staging directory. This will be referred to as STAGING_DIR for database installation tasks.
3. Change directories to STAGING_DIR and extract the ari14dbserverunix.zip file.

Create Staging Directory for ARI Application Files

1. Create a staging directory for the ARI application software. There should be a minimum of 50 MB disk space available for the application installation files.
2. Copy the file ari14appserverunix.zip from the CDROM directory to the staging directory. This is referred to as STAGING_DIR for application installation tasks.
3. Change directories to STAGING_DIR and extract the file ari14appserverunix.zip.
4. Confirm that all scripts in STAGING_DIR/forms11g_scripts have at least execute permissions for the oretail user and its group (r-xr-x---).

A Note to Retailers Using the Croatian Language

See My Oracle Support document ID #393320.1 for important information regarding steps to enable the Croatian language for Forms and Reports.

2

RAC and Clustering

Oracle Retail Active Retail Intelligence has been validated to run in two configurations on Linux:

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

The Oracle Retail products have been validated against a 12.1.0.1 RAC database. When using a RAC database, all JDBC connections should be configured to use THIN connections rather than OCI connections. It is suggested that 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 10.3.6 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 12.1.0.1 Oracle Internet Directory database with the WebLogic 10.3.6 cluster. It is suggested that a Web Tier 11.1.1.7 installation be configured to reflect all application server installations if SSO will be utilized. References for Configuration:

- Oracle® Fusion Middleware High Availability Guide 11g Release 1 (11.1.1) Part Number E10106-09
- Oracle Real Application Clusters Administration and Deployment Guide 12c Release 1 (12.1) E48838-08

3

Database Installation Tasks

It is assumed that Oracle 12c release1, with appropriate patches, has already been installed. If not, refer to "*Check Supported Database Server Requirements*" in Chapter 1, "Preinstallation Tasks" before proceeding**.** Additionally, STAGING_DIR in this section refers to the directory created in "Create Staging Directory for ARI Database Files", Chapter 1.

Although ARI can exist as a standalone application, these directions assume that it will be installed in an existing RMS database. If this is not the case, it is necessary to create a database per information in the section "Create the Database as Follows". Refer to Appendix A for additional information. Once that has been completed complete the remaining steps in this section.

If ARI will be installed in an existing RMS database proceed to section "Verify Existence of Oracle Packages" and complete the remaining steps.

Note: When running the scripts in this section, the following errors may be encountered:

ORA-00942: table or view does not exist
ORA-00955: name is already used by an existing object
ORA-01432: public synonym to be dropped does not exist
ORA-01434: private synonym to be dropped does not exist
ORA-01921: role name 'XXXXXXX' conflicts with another user or role
ORA-02289: sequence does not exist
ORA-04042: procedure, function, package, or package body does not exist
ORA-04043: object XXXXXXX does not exist
ORA-29807: specified operator does not exist
ORA-29833: indextype does not exist
ORA-29931: specified association does not exist

These errors can be ignored. The ORA errors are caused by dropping the objects the script is about to create.

Create the Database Instance Using Oracle Generic Template

Prerequisites:

- 12.1.0.1 binary must have already been installed along with 12.1.0.1.4 patchset. Refer to the Database Server Preinstallation section for all the required oneoff patches.

Background

As of 14.1, Oracle Retail no longer delivers customed database template files. Instead, databases can be created using the generic Oracle delivered template in the directory: $ORACLE_HOME/assistant/dbca/template.

```
msp52542:[polsc01] /u00/oracle/product/12.1.0.1/assistants/dbca/templates>
--> ls -l General_Purpose.dbc
-rw-r--r-- 1 oracle rgbudba 4908 May 24  2013 General_Purpose.dbc
```

Instance Creation Using the Generic Template via DBCA

1. Ensure ORACLE_HOME and ORACLE_BASE is in the path:

   ..export ORACLE_HOME=/u00/oracle/product/12.1.0.1

   ..export ORACLE_BASE=/u00/oracle

   .. export PATH=$ORACLE_HOME/bin:$PATH

   .. cd into /u00/oracle/product/12.1.0.1/assistants/dbca/templates

2. Execute the following command to create an instance:

   ```
   $ORACLE_HOME/bin/dbca -silent -createDatabase -templateName
   General_Purpose.dbc -gdbName DB_NAME -sid DB_SID  -
   createAsContainerDatabase true -SysPassword oracle1 -SystemPassword
   oracle1 -emConfiguration NONE -datafileDestination /u02/oradata -
   characterSet AL32UTF8 -nationalCharacterSet AL16UTF16 -
   redoLogFileSize 100 -initParams nls_date_format=DD-MON-
   RR,nls_language=AMERICAN,nls_calendar=GREGORIAN,fast_start_mttr_targe
   t=900
   ```

   The above will create a container database using all the default parameters set by dbca. Please replace the pfile by taking a copy from Appendix A but customize the values according to the need of your environment.

   If you wish to create a non-container database, replace [-createAsContainerDatabase true] with [-createAsContainerDatabase false].

3. Execute the following command to create a pluggable database if this is a container environment

   ```
   CREATE PLUGGABLE DATABASE PDB_NAME ADMIN USER PDBADMIN
   IDENTIFIED BY pdbadmin_pwd ROLES=(CONNECT)
   file_name_convert=('/u02/oradata/cdb_name/pdbseed','/u02/oradata/pdb_name');

   alter pluggable database pdb_name open;

   alter system register;
   ```

4. Post Database Creation Setup

   The above commands create a database with all files in one directory, ie, /u02. Please multiplex the redo logs and the controlfiles following the OFA architecture.

5. Configure the listener and the tnsnames entry.
6. Log into the pluggable database to create the required tablespaces accordingly. For non-container databases, log into the database as normal to create the tablespaces.

Verify the Existence of Oracle Packages

Confirm that the DBMS_SESSION, DBMS_RANDOM, DBMS_ALERT, DBMS_PIPE, DBMS_JOB packages exist. As sysdba, run the following query:

```
SQL>    select object_name
        from dba_objects
        where owner='SYS'
        and object_name in ('DBMS_SESSION','DBMS_RANDOM',
        'DBMS_ALERT','DBMS_PIPE','DBMS_JOB');
```

The source for these packages are located in the $ORACLE_HOME/rdbms/admin directory. If they do not exist, create them by executing

```
@$ORACLE_HOME/rdbms/admin/catproc.sql as sysdba.
```

Create ARI Tablespaces

Two tablespaces named ari_data and ari_index are required.

1. Modify STAGING_DIR/ari/create_db/create_ari_tablespaces.sql. Refer to comments in this file regarding modifications that need to be made.
2. Login to SQL*Plus as sysdba and execute create_ari_tablespaces.sql.

Create ARI Schemas

1. Create a schema that owns the ARI database objects. The following script prompts for the schema name and password. A suggested name for this schema is 'ARI141M'. This is referred to as the <master schema owner>.
2. Change directories to STAGING_DIR/ari/utility
3. Log into SQL*Plus as sysdba and execute the following:
   `create_master_schema_user.sql`
4. Create a schema that will be used for ARI-generated trigger, packages, procedures and tables. The following script prompts for the schema name and password. A suggested name for this schema is 'ARI141G'. This will be referred to as the <generated schema owner>.
5. Change directories to STAGING_DIR/ari/utility
6. Log into SQL*Plus as sysdba and execute the following:
   `create_gen_schema_user.sql`

Set Environment Variables

The following variables must be set before installing ARI.

Variable:	Description:
ORACLE_HOME	The database location where your Oracle Retail application will be installed.
ORACLE_SID	Example: ORACLE_SID=retaildb
NLS_LANG	The locale setting for Oracle database client. Example: NLS_LANG=AMERICAN_AMERICA.AL32UTF8
ORACLE_INSTANCE	Location where WebLogic has been installed and is where you will find the bin dir which holds the frmcmp executable. Example: /u00/webadmin/product/10.3.6/WLS/OracleFR_1

Create ARI Database Objects

Note: When running the scripts in this section the following errors may be encountered "Warning: View created with compilation errors" or "Warning: Package created with compilation errors". These errors can be ignored. The warnings are caused by dependencies on objects that get created later in the install. The warnings will be cleared when objects are re-validated towards the end of the database install.

1. Change directories to STAGING_DIR/ari/ddl/source.
2. Log into SQL*Plus as <master schema owner> and execute ari14.sql. Review ari14.log for errors and correct as needed.
3. Change directories to STAGING_DIR/ari/packages/source.
4. Log into SQL*Plus as <master schema owner> and execute ari14dbo.sql. Review ari14dbo.log for errors and correct as needed.
5. Log into SQL*Plus <master schema owner> as and execute STAGING_DIR/ari/utility/inv_obj_comp.sql to validate any objects that may be invalid.

Create ARI Data

1. Change directories to STAGING_DIR/ari/data/source.
2. Log into SQL*Plus as <master schema owner> and execute ari14ctl.sql. This script calls several scripts one of which is ari_options.sql. When prompted, enter values for the master and generated schema names in upper case when indicated.
3. Upon completion, check the spool file, ari14ctl.log, to verify that no errors were received.
4. Change directories to STAGING_DIR/ari/control_scripts/source.
5. Log into SQL*Plus as <master schema owner> and execute ari14scripts.sql.
6. Upon completion, check the spool file, ari14scripts.log, to verify that no errors were received.
7. Change directories to STAGING_DIR/ari/data/source/form_menu_elements.
8. Log into SQL*Plus as <master schema owner> and run the following command:
   ```
   SQL> @base_form_menu_elements.sql
   ```

Insert Language Data

Insert Secondary Language Data

> **Note:** These scripts are only for customers who wish to have a primary language of English and a secondary language of any combination of released languages.

1. Change directories to STAGING_DIR/ari/data/source/forms_menu_elements
2. Log into sqlplus as ARI 14 MASTER and run the following command:
   ```
   SQL> @base_form_menu_elements_langs_<lang>.sql (where <lang> is the language code)
   ```
   Language codes are as follows:
   - de – German
   - es – Spanish
   - fr – French
   - ja – Japanese
   - ko – Korean
   - it – Italian
   - ru - Russian
   - ptb – Brazilian Portuguese
   - zhs – Simplified Chinese
   - zht – Traditional Chinese
   - el – Greek
   - hr – Croatian
   - hu – Hungarian
   - nl – Dutch
   - pl – Polish
   - sv – Swedish
   - tr - Turkish

Note: If other languages are desired, please use these same steps substituting the language, <lang>.

Insert Primary Language Data

Note: These scripts are only for customers who wish to have a primary language other than English. Secondary language support is not possible if the primary language is other than English.

1. Change directories to STAGING_DIR/ari/data/source/data_<lang>
2. Set the SQL*Plus session so that the characterset component of the NLS_LANG is AL32UTF8.

   Example: RUSSIAN_RUSSIA.AL32UTF8
3. Log into SQL*Plus as <master schema owner> and run the following command:
   ```
   SQL> @ ari14_[lang].sql
   ```
4. Check the log file ari14_[lang].log for any errors.
5. Change directories to STAGING_DIR/ari/data/source/forms_menu_elements
6. Log into SQL*Plus as <master schema owner> and run the following command:
   ```
   SQL> @base_form_menu_elements_langs_[lang].sql
   ```

Alter ARI Data

1. Change directories to STAGING_DIR/ari/interface/source.
2. Log into SQL*Plus as <master schema owner> and run the following commands:

   SQL> @disable_mts_fks.sql

   SQL> @mts_realm.sql

   SQL> @mts_parm_type.sql

   SQL> @mts_parm.sql

   SQL> @update_group_lookup.sql

   SQL> @enable_mts_fks.sql

Create Generated Schema Synonyms

This script prompts for values for the master schema and generated schema names.

1. Change directories to STAGING_DIR/ari/utility.
2. Log into SQL*Plus as <generated schema owner> and run the following command:
   SQL> @generated_syns.sql

Revoke Installation-only Privileges

Certain master and generated schema system privileges are only required during the installation process. Create session and create synonym can be revoked from the generated schema. Create sequence and create view can be revoked from the master schema. This script prompts for values for the master and generated schema names.

1. Change directories to STAGING_DIR/ari/utility.
2. Log into SQL*Plus as sys and run the following script:
   ```
   SQL> @revoke_install_privs.sql
   ```

Create User Synonyms

Each additional user schema of ARI 14.1 requires synonyms to the <master schema owner> objects and the product(s) that it will be integrated with (e.g. RMS). After ensuring that each new user has the 'create synonym' system privilege, create synonyms to each <master schema owner> object of type table, view, function, package, procedure or sequence. The user_syns.sql script prompts for values for the master schema name, the user name, password and database of the user you are creating the synonyms for. This script will create the synonyms to the <master schema owner> objects.

1. Change directories to STAGING_DIR/ari/utility.
2. Log into SQL*Plus as <master schema owner> and run the following script:

```
SQL> @user_syns.sql
```

Integration with RMS

To configure the integration of ARI and RMS, the RMS schema will need synonyms to the ARI schema.

1. Log into SQL *Plus as <RMS Schema Owner>

```
SQL>  drop package ARI_INTERFACE_SQL;
```

2. Log into SQL *Plus as <master schema owner>

```
SQL> @user_syns.sql
```

3. When prompted for the user name, use <RMS Schema Owner>.

Configuring ARI for Email Alerts

To configure ARI for email alerts, the ARI_OPTIONS table must be updated, the UTL_MAIL package needs to be created, and an access control list configured.

1. Log into SQL *Plus as <master schema owner>.
2. Run the below SQL, replacing the option_value with valid email address and mail routing server name

```
UPDATE ARI_OPTIONS SET OPTION_VALUE = '<valid email address>' WHERE
OPTION_NAME = 'MAIL_SERVICE_ADDRESS';
UPDATE ARI_OPTIONS SET OPTION_VALUE = '<valid mail routing server>' WHERE
OPTION_NAME = 'MAIL_HOST';
COMMIT;
```

3. Log into SQL *Plus as sys.
4. Create the UTL_MAIL and UTL_MAIL_INTERNAL packages.

```
SQL> @$ORACLE_HOME/rdbms/admin/utlmail.sql
SQL> @$ORACLE_HOME/rdbms/admin/prvtmail.plb
```

Create an access control list to use with UTL_MAIL

1. Change directories into STAGING_DIR/ari/utility
2. Log into SQL *Plus as sys.

```
SQL> @tcp_acl_util.sql
```

This script will read values from ARI_OPTIONS table needed to create the ACL.

```
SQL> GRANT EXECUTE ON UTL_MAIL TO <master schema owner>;
```

3. To test the setup, log into SQL *Plus as <master schema owner>.

```
begin
 UTL_MAIL.send(sender => '<valid from email address>',
    recipients => '<valid to email address>',
     subject => 'Email testing',
     message => 'This is a test email for ARI',
     mime_type => 'text; charset=us-ascii');
end;
/
```

Note: If you get the below error while trying to test using UTL_MAIL is successful, you will need to set the smtp_out_server value for the system.

ERROR at line 1:

ORA-06502: PL/SQL: numeric or value

ORA-06512: at "SYS.UTL_MAIL", line 654

As SYS user, run:

alter system set smtp_out_server = '<valid mail routing erver>;

4

Application Installation Tasks

These instructions assume that WebLogic 10.3.6 has been already been installed. If not, refer to "*Check Application Server Requirements*" in Preinstallation Tasks before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in Create Staging Directory for ARI Application Files.

Installation Preparation

1. Logon to the application server as the oretail user.
2. Create a directory that will hold the installed ARI forms.

   For example: mkdir /u00/oretail/ari

   > **Note:** This directory will be referred to as INSTALL_DIR for the remainder of the document.

3. Set and export the following variables:

Variable	Description	Example
NLS_LANG	Locale setting for Oracle database client.	export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
WLS_HOME	Point to your Weblogic installation	export WLS_HOME=/u00/webadmin/product/10.3.6/WLS
ORACLE_HOME	Point to your Forms & Reports software installation	export ORACLE_HOME=$WLS_HOME/Oracle_FRHome1
ORACLE_INSTANCE	Points to the instance of Forms & Reports	export ORACLE_INSTANCE=$WLS_HOME/asinst_1
ORACLE_SID	The database/SID where the ARI schema resides.	export ORACLE_SID=retaildb
DISPLAY	Address and port of X server on desktop system of user running install. Required for forms application installer.	export DISPLAY=<IP address>:0

4. The T2kMotif.rgb file that is sent out with WebLogic (10.3.6) must be modified. It is located at the following location:

   $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
5. Make a copy of the file Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).
6. Modify the file Tk2Motif.rgb file so that it contains the following line:

   ```
   Tk2Motif*fontMapCs: iso8859-2=AL32UTF8
   ```
7. Set and export the following variables:
   - PATH=$ORACLE_INSTANCE/bin:$ORACLE_HOME/bin:$ORACLE_HOME/opmn/bin:STAGING_DIR/forms11g_scripts:$PATH

- CLASSPATH=$ORACLE_HOME/jlib/importer:$ORACLE_HOME/jlib/debugger.jar:$ORACLE_HOME/jlib/utj.jar:$ORACLE_HOME/jlib/ewt3.jar:$ORACLE_HOME/jlib/share.jar:$ORACLE_HOME/jlib/dfc.jar:$ORACLE_HOME/jlib/help4.jar:$ORACLE_HOME/jlib/oracle_ice.jar:$ORACLE_HOME/jlib/jewt4.jar
- FORMS_BUILDER_CLASSPATH=$CLASSPATH
- FORMS_PATH= INSTALL_DIR/forms/bin:$ORACLE_HOME/forms
- TK_UNKNOWN=$ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
- UP=<ARI master schema owner>/<ARI master schema password>@<ARI database>
- Or using the wallet UP=/@<alias>

Note: Verify that TNS is set up correctly by using the UP variable to successfully log into the ARI 14 schema.

Example: /u00/oretail> sqlplus $UP

ARI Forms Installation

1. Create the forms/src directory under INSTALL_DIR.
   `mkdir –p INSTALL_DIR/forms/src`
2. Create the forms/bin directory under INSTALL_DIR.
   `mkdir –p INSTALL_DIR/forms/bin`
3. Copy all files from STAGING_DIR/forms/bin to INSTALL_DIR/forms/bin.
   `cp STAGING_DIR/forms/bin/* INSTALL_DIR/forms/bin`
4. Copy all files from STAGING_DIR/forms/src to INSTALL_DIR/forms/src.
   `cp STAGING_DIR/forms/src/* INSTALL_DIR/forms/src`
5. Copy all libraries (*.pll files) in the INSTALL_DIR/forms/src directory to the directories to the INSTALL_DIR/forms/bin directory.
6. Change directories to INSTALL_DIR/forms/bin.
7. Run pll2plx11g_ari to compile all ARI .pll's.

   Note: If the pll2plx11g_ari script is not used and the libraries are compiled individually, then they must be compiled in the following order (which is noted in the pll2plx11g_ari script):

   - ariiflib90.pll
   - arimessage.pll
   - ariswidget.pll
   - aristandard.pll
   - arimblock.pll
   - arimview.pll
8. Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully).
9. Remove all newly created .plx files.

10. Copy all forms (*.fmb files) in the INSTALL_DIR/forms/src directory to the INSTALL_DIR/forms/bin directory.
11. Run fmb2fmx11g_fm (in INSTALL_DIR/forms/bin) to compile the ARI reference forms.

> **Note:** If the fmb2fmx11g_fm script is not used and the libraries are compiled individually, then they must be compiled in the following order:

- fm_refer
- fm_date
- fm_edit
- fm_mblk
- fm_multi
- fm_work
- fm_xtet

12. Check to make sure that each reference form (fm_*.fmb) file has been compiled by verifying the time stamp changed. fm_edit, fm_multi, and fm_work will not generate an .fmx file which is fine.

> **Note:** Disregard fm_*.fmx files should they be created. These files should be removed.

13. Remove all newly created fm_*.fmx files (reference forms should not have executable files).
14. Run fmb2fmx11g (in INSTALL_DIR/forms/bin) to generate ARI runtime forms – .fmx's.
15. Check to make sure that each non-reference form (.fmb file) has a corresponding .fmx (to ensure that all non-reference .fmb's compiled successfully).
16. Remove all non-reference form forms from INSTALL_DIR/forms/bin; the following syntax leaves all reference forms (fm_*.fmb) in the bin directory, while removing all other forms:

```
> for PROG in `ls *.fmb | grep -v fm_`
> do PROGNAME=`echo $PROG`
> rm $PROGNAME
> done
```

17. Copy all menus (*.mmb files) in the INSTALL_DIR/forms/src directory to the INSTALL_DIR/forms/bin directory.
18. Run mmb2mmx11g (in INSTALL_DIR/forms/bin) to generate ARI runtime menus – .mmx's.
19. Check to make sure that each .mmb file has a corresponding .mmx file.
20. Remove all .mmb files from INSTALL_DIR/forms/bin.

> **Note:** .err files may be created by the compilation scripts above. These files are logs of the compilation process and can be removed.

Configure WebLogic 10.3.6 for ARI

> **Note:** The proper WebLogic 10.3.6 components must be started in order to run Forms applications.
>
> **Note:** WLS_HOME refers to the location where WebLogic 10.3.6 is installed.

1. Make a copy of the file $WLS_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/default.env, and name it ari.env (for example).
2. Modify the new file ari.env by appending the location of the ARI forms modules to the FORMS_PATH variable setting, and by adding the NLS_DATE_FORMAT and NLS_LANG variables to the end of this file. Additionally, the variable FORMS_REJECT_GO_DISABLED_ITEM=FALSE and FORMS_USERNAME_CASESENSITIVE=1 must also be added to ari.env.

   Example:
   FORMS_PATH=/u00/oretail/ari/forms/bin:/u00/oracle/product/10.3.6/WLS/OracleFR_1/FormsComponent/forms
   NLS_DATE_FORMAT=DD-MON-RR

   NLS_LANG=AMERICAN_AMERICA.AL32UTF8

   FORMS_USERNAME_CASESENSITIVE=1
   FORMS_REJECT_GO_DISABLED_ITEM=FALSE

3. Modify the file formsweb.cfg located at $WLS_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/ by creating the ARI environment section at the end of this file. Brackets ([ ] in the example below) distinguish a separate environment in this file. Variables to be set in the ARI environment section of formsweb.cfg are: envfile (from step 2 above); width, height, and separateFrame applet parameters; and starting form for the ARI application.

   Example:
   [ari]
   envfile=ari.env
   width=850
   height=585
   separateFrame=true

   lookAndFeel=Oracle

   colorScheme=swan

   archive=frmall.jar,ari-icons.jar

   imageBase=codeBase

   form=arimstr.fmx

   If Oracle Single Sign-On is to be used with ARI, then

   - Set ssoMode to webgate.

- If Resource Access Descriptors are allowed to be dynamically created, then set ssoDynamicResourceCreate to true.

Example: [ari]

envfile=ari.env
width=850
height=585
separateFrame=true

lookAndFeel=Oracle

colorScheme=swan

archive=frmall.jar,icons.jar

form=arimstr.fmx

ssoMode=webgate

ssoDynamicResourceCreate=true

4. Copy ARI icons to $ORACLE_HOME/forms/java.
   `cp STAGING_DIR/web_html/ari-icons.jar $ORACLE_HOME/forms/java`
5. Update the Forms Registry.dat file with where the icons are to be fetched: $WLS_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/forms/registry/oracle/forms/registry/Registry.dat

   default.icons.iconpath=
   default.icons.iconextension=gif
6. When integrating ARI with RMS, update the FORMS_PATH in the RMS env file to include the ARI forms modules in addition to the RMS forms modules.
7. Launch ARI by entering the following URL in a browser. Prior to testing, the Sun JRE 1.8 plug-in needs to be installed on the client machine. The plug-in can be downloaded from http://java.sun.com/.

   http://<server>:<port>/forms/frmservlet?config=<env>

   - server = name or IP address of server where Oracle Forms Services 11g Release 2 (11.1.2.2) is running
   - port = "Listen" value in ORACLE_HOME/user_projects/domains/ClassicDomain/config/config.xml It is the port listed for WLS_FORMS
   - env = name of the environment in brackets in formsweb.cfg (from step 3 above).

Note: If ARI is configured to use SSO (ssoMode = webgate), then the Oracle Single Sign-On page should appear. Login using a valid user ID / password found in the OID LDAP server.

8. If Single Sign-On is not used, or if a Resource Access Descriptor has not been set up for ARI for this user and ssoDynamicResourceCreate is true, then the ARI logon form appears. On the ARI logon form, enter the appropriate *Username/Password@Connect String* information in the corresponding fields:
   - Username = ARI Master Schema Owner or additional Oracle user created
   - Password = Username password
   - Connect String = Oracle database

   Example: Username: ARI141M
   Password: retail
   Connect String: prod_db1

Helpfile Installation

1. Log into the WebLogic Admin console to which online help will be installed.
2. Create a server. In this example **ari-help** is being used.
3. On the left in the console, select Environment/Servers.
4. Click **Lock & Edit**.
5. Click **New**.
   a. Enter the managed server name ari-help.
   b. Leave Server Listen Address blank.
   c. Enter an unused Listen Port (for example 7008).
   d. Choose No cluster.
   e. Click **Finish**.
6. Click **Activate Changes**.
7. Click the Configuration tab and click on the server.
   a. Click **Lock & Edit**.
   b. Change the Machine dropdown to the node manager machine.
   c. Click **Save**.
8. Click **Activate Changes**.
9. Go back to the Summary of Servers page.
10. Click the Control tab and check the server. Click **Start**.
    Wait for the ari-help server to change state to "Running"
11. Click **Deployments** on the left.
12. Click **Lock & Edit**.
13. Click the **Install** button.
    a. Change the path to STAGING_DIR/ari/online-help.
    b. Select ari-help.ear.
    c. Click **Next**.
    d. Select **Install this Deployment as an Application**.
    e. Click **Next**.
    f. Select managed server, ari-help, that was created earlier.
    g. Select remaining defaults and click **Finish**.
14. Click **Activate Changes**.
15. Select deployment and click Start->Servicing all requests.

16. Log into sqlplus as the ARI 14 master schema owner (ARI141M) and update the ari_language table so that WEBHELP_SERVER is correct:

    WEBHELP_SERVER is the URL http://<server>:<port>**.** where <server> is the name or IP address of the server where WebLogic is installed and <port> is the value set when the ari-help-server was created.

    Example: SQL> update ari_language set WEBHELP_SERVER='http://server:7008' where lang=1;

Import-Export Tool Installation Instructions

The current version of IET (ARI Import-Export Tool) is 1.3.1 (provided in the IET directory). Most clients want to install IET so that they can import prepackaged rules, and move rules between ARI instances. The IET Windows Installer is the file ariiet131.exe. Run this installer on the Windows machine that you want to run IET on (should have database access to all ARI instances). Follow the directions within the installer to complete your IET installation. IET requires a JDK 1.3 compliant Java Virtual Machine; the installer gives you the option of using an existing JVM or installing one that is bundled with IET.

A

Appendix: Oracle 12cR1 Database Parameter File

```
###########################################################################
# Copyright (c) 2014 by Oracle Corporation
# Oracle 12.1.0.x Parameter file
# NOTES: Before using this script:
#        1.  Change <datafile_path>, <admin_path>, <utl_file_path>, <diag_path>
and <hostname>
#            values as appropriate.
#        2.  Replace the word SID with the database name.
#        3.  Size parameters as necessary for development, test, and production
environments.
# ---------------------------------------------------------------------
*.audit_file_dest=full_path_of_audit_dir
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='full_path_of_controlfile_1','full_path_of_controlfile_2'
###############################################
# Memory Settings:
# xxxM = Some reasonable starting value for your environment.
###############################################
*.db_block_size=xxxM
*.db_cache_size=xxxM
*.java_pool_size=xxxM
*.memory_target=xxxM
*.pga_aggregate_target=xxxM
*.shared_pool_size=xxxM
*.streams_pool_size=xxxM

###############################################

*.db_block_size=8192
*.db_domain=''
*.db_name='dbName'
*.diagnostic_dest='full_path_of_diag_dir'
*.enable_pluggable_database=true|false
*.fast_start_mttr_target=900
*.nls_calendar='GREGORIAN'
*.nls_date_format='DD-MON-RR'
*.nls_language='AMERICAN'
*.nls_numeric_characters='.,'
*.nls_sort=BINARY
*.open_cursors=900
*.os_authent_prefix=''
*.plsql_optimize_level=2
*.processes=2000
*.query_rewrite_enabled='true'
*.remote_dependencies_mode='SIGNATURE'
*.remote_login_passwordfile='EXCLUSIVE'
*.remote_os_authent=true
*.sec_case_sensitive_logon=false
*.undo_tablespace='UNDOTBS1'
```

B

Appendix: Create ARI Tablespaces

```
--------------------------------------------------------------------------------
---
--- Script:        create_ari_tablespaces.sql
--- Execute as:    sysdba
--- Note:          Before running this script:
---                  Modify <datafile_path> values.
---                  Modify datafile storage parameters and sizes as needed
--------------------------------------------------------------------------------
spool create_ari_tablespaces.log

CREATE TABLESPACE ARI_INDEX
DATAFILE '<datafile_path>/ari_index01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE ARI_DATA
DATAFILE '<datafile_path>/ari_data01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;

spool off
exit
```

C

Appendix: Single Sign-On for WebLogic

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

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

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

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

What Do I Need for Single Sign-On?

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

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle Access Manager (OAM) 11g Release 2 server and administrative console for implementing and configuring policies for single sign-on.
- A Policy Enforcement Agent such as Oracle Access Manager 11g Agent (WebGate), used to authenticate the user and create the Single Sign-On cookies.
- Oracle Directory Services Manager (ODSM) application in OIM11g, used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OAM system and registering HTTP servers.

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

Can Oracle Access Manager Work with Other SSO Implementations?

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

Oracle Single Sign-on Terms and Definitions

The following terms apply to single sign-on.

Authentication

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

Dynamically Protected URLs

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

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

Oracle Identity Management (OIM) 11g includes Oracle Internet Directory and ODSM. Oracle Access Manager (OAM) 11g R2 should be used for SSO using WebGate. Oracle Forms 11g contains Oracle HTTP server and other Retail Applications will use Oracle WebTier11g for HTTP Server.

MOD_WEBLOGIC

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

Oracle Access Manager 11g Agent (WebGate)

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

Oracle Internet Directory

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

Partner Application

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

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

Statically Protected URLs

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

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

What Single Sign-On is not

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

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

How Oracle Single Sign-On Works

Oracle Access Manager involves several different components. These are:

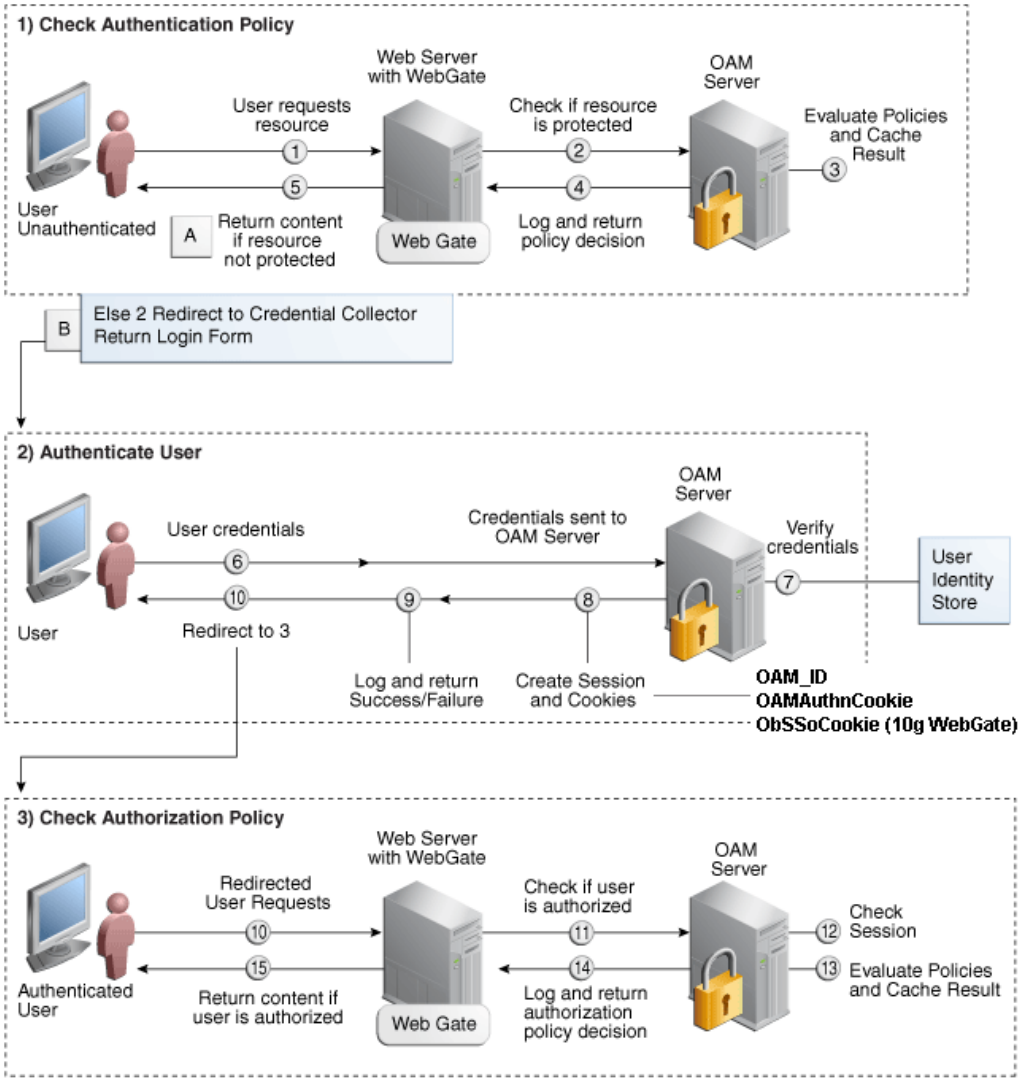
- The Oracle Access Manager (OAM) server, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle Access Manager Agent associated with the Web application, which verifies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

About SSO Login Processing with OAM Agents

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

15. WebGate responds as follows:
    - If the authorization policy allows access, the desired content or applications are served to the user.
    - If the authorization policy denies access, the user is redirected to another URL determined by the administrator.

SSO Login Processing with OAM Agents



Installation Overview

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

1. Oracle Internet Directory (OID) LDAP server and the Oracle Directory Services Manager. They are typically installed using the Installer of Oracle Identity Management . The ODSM application can be used for user and realm management within OID.
2. Oracle Access Manager 11gR2 has to be installed and configured.
3. Additional midtier instances (such as Oracle Forms 11gr2) for Oracle Retail applications based on Oracle Forms technologies (such as RMS). These instances must be registered with the OAM installed in step 2.
4. Additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities must be registered with OAM installed in step 2.

Infrastructure Installation and Configuration

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

OID User Data

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

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

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

User Management

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

ODSM

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

LDIF Scripts

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

User Data Synchronization

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

D

Appendix: Single Sign-On Resource Access Descriptors

Oracle Forms applications such as ARI use database connections for authentication and authorization purposes. Oracle Single Sign-On, however, uses the Oracle Internet Directory (OID) user ID and password for this purpose. The Forms framework maps OID user IDs to database connections via information stored in Resource Access Descriptors (RADs). A user will have one RAD for each application accessed. RADs may be created by an administrator or by an LDIF script. Depending on the Oracle Internet Directory and/or the formsweb.cfg configuration, RADs may also be created by the user.

A user is prompted for the database connection information whenever formsweb.cfg file specifies ssoMode = true and createDynamicResources = true for an application and no valid RAD exists. RADs may become invalid when passwords have expired or have been changed.

RADs may be created by administrators or users via the Delegated Administration Services application.

> **Note:** Users can create new RADs only if one or more RADs already exist.

RADs may be created and via LDIF scripts as well. Documentation on this may be found in the My Oracle Support document number 244526.1.

E

Appendix: Common Errors

FRM -93552: cannot connect to runtime process. Error when using ARI in a SSO environment

Symptom

When launching multiple applications in a SSO environment, ARI forms can fail with:

```
FRM-93552: cannot connect to runtime process.
```

Solution

You need to change the default JSESSIONID cookie name for the forms process. There are two articles from Oracle Support that document this process:

- How to Change the Default JSESSIONID Cookie Name for Forms (Doc ID 1578506.1)
- How To Redeploy the Forms Application after Modification of Forms J2EE Application Deployment Descriptors (Doc ID 1063045.1)

F

Appendix: Setting Up Password Stores with wallets/credential stores

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

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

ORACLE Retail Merchandising applications now have 3 different types of password stores. They are database wallets, java wallets, and database credential stores. Background and how to administer them below are explained in this appendix

About Database Password Stores and Oracle Wallet

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

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

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

```
sqlplus /@db_username
```

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

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

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

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

Setting Up Password Stores for Database User Accounts

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

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

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

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

1. Create a wallet using the following command:

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

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

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

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

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

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

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

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

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

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

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

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

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

Setting up Wallets for Database User Accounts

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

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

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

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

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

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

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

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

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

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

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

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

Examples for a NON pluggable db:
dvols29_rms01user =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = xxxxxx.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = <sid_name> (GLOBAL_NAME = <sid_name>)))

dvols29_rms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = xxxxxx.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = <sid_name>) (GLOBAL_NAME = <sid_name>)))

Examples for a pluggable db:
dvols29_rms01user =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = xxxxxx.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = <pluggable db name>)))

dvols29_rms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = xxxxxx.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = <pluggable db name>)))
```

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

4. Create the wallet files. These are empty initially.
   a. Ensure you are in the intended location.
      ```
      $ pwd
      /projects/rms14/dev/.wallet
      ```
   b. Create the wallet files.
      ```
      $ mkstore -wrl . –create
      ```
   c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.
   d. Enter the password again.

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

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

6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.
   ```
   $ export TNS_ADMIN=/projects/rms14/dev/.wallet /* This is very import to use wallet to point at the alternate tnsnames.ora created in this example */

   $ sqlplus /@dvols29_rms01user

   SQL*Plus: Release 12

   Connected to:
   Oracle Database 12g

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

   Running batch programs or shell scripts would be similar:

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

   Set the UP unix variable to help with some compiles :

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

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

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

```
mkstore –wrl . –deleteCredential dvols29_rms01user
```

- Change the password for a credential on wallet

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

- List the wallet credential entries

```
mkstore –wrl . –list
```

This command returns values such as the following.

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

- View the details of a wallet entry

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

Returns the value of the entry:

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

Returns the value of the entry:

```
rms01user
```

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

Returns the value of the entry:

```
Passwd
```

Setting up RETL Wallets

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

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

To set up RETL wallets, perform the following steps:

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

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

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

   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     - `export RETL_WALLET_ALIAS="retl_java_rms01user"`
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     - `export ORACLE_WALLET_ALIAS="dvols29_rms01user"`
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     - `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"`
5. To change a password later, run `setup-security-credential.sh`.
   - Enter 2 to update a database credential.
   - Select the credential to update.
   - Enter the database user to update or change.
   - Enter the password of the database user.
   - Re-enter the password.

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

For Java applications, consider the following:

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

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

- Two main scripts are related to this script in the folder for more generic wallet operations: dump_credentials.sh and save_credential.sh.

- If you have not installed the application yet, you can unzip the application zip file and view these scripts in <app>/application/retail-public-security-api/bin.
- Example:
- /u00/webadmin/reim14/application/retail-public-security-api/bin

update-<ALIAS>.sh

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

Usage:

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

Example:

```
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/reta
il-public-security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass in.
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/reta
il-public-security-api/bin>
```

dump_credentials.sh

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

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

Example:

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

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

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

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

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

save_credential.sh

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

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

Example:

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

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

Enter password:
Verify password:
```

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

Usage

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

How does the Wallet Relate to the Application?

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

Reim.properties code sample:

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

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

How does the Wallet Relate to Java Batch Program use?

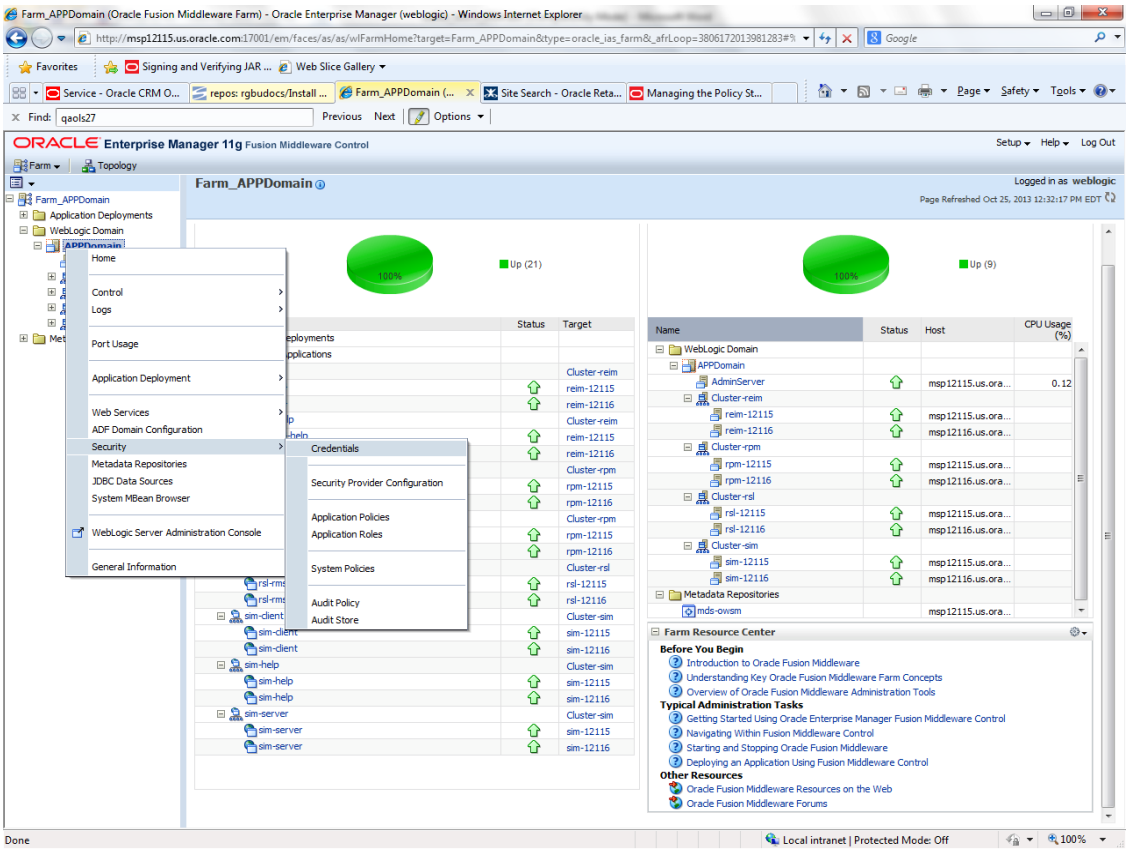
Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser RMS01APP, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments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

Database Credential Store Administration

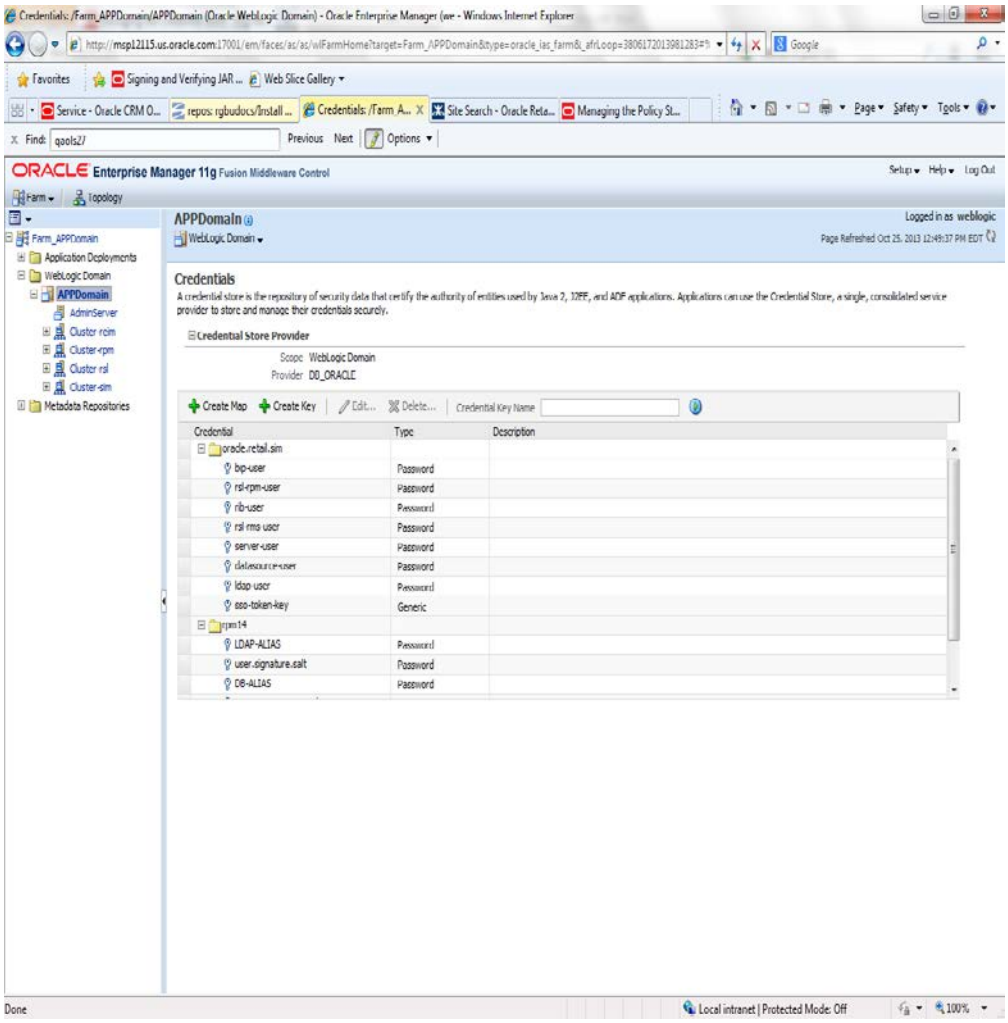
The following section describes a domain level database credential store. This is used in RPM login processing, SIM login processing, RWMS login processing, RESA login processing and Allocation login processing and policy information for application permission. Setting up the database credential store is addressed in the RPM, SIM, RESA, RWMS, and Alloc 14.1 install guides.

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

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

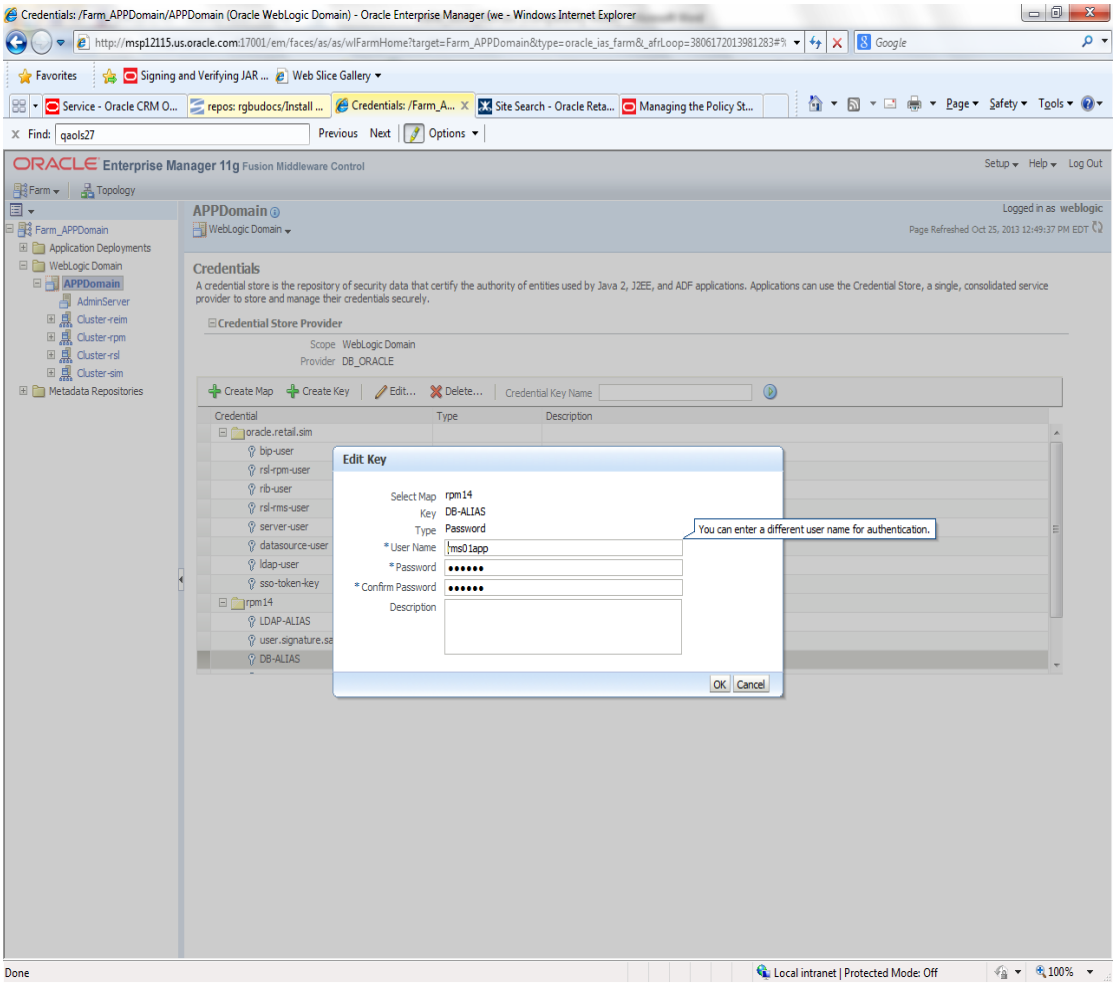


2. Click on Credentials and you will get a screen similar to the following. The following screen is expanded to make it make more sense. From here you can administer credentials.


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

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

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



The screen above shows the map (rpm14) that came from the application installer, the key (DB-ALIAS) that came from the application installer (some of the keys/alias are selected by the person who did the application install, some are hard coded by the application installer in question), the type (in this case password), and the user name and password. This is where you would check to see that the user name is correct and reset the password if needed. REMEMBER, a change to an item like a database password WILL make you come into this and also change the password. Otherwise your application will NOT work correctly.

Managing Credentials with WSLT/OPSS Scripts

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

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

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

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

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

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

/u00/webadmin/product/wls_apps/oracle_common/bin> ./orapki wallet display –wallet /u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig

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

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

Oracle PKI Tool : Version 11.1.1.7.0

Requested Certificates:

User Certificates:

Oracle Secret Store entries:

rpm14@#3#@DB-ALIAS

rpm14@#3#@LDAP-ALIAS

rpm14@#3#@RETAIL.USER

rpm14@#3#@user.signature.salt

rpm14@#3#@user.signature.secretkey

rpm14@#3#@WEBLOGIC-ALIAS

rpm14@#3#@WLS-ALIAS

Trusted Certificates:

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

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

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

listCred

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

Script Mode Syntax

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

Interactive Mode Syntax

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

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

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

Examples of Use:

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

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

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

```
/u00/webadmin/product/wls_apps/oracle_common/common/bin>
sh wlst.sh

Initializing WebLogic Scripting Tool (WLST)...

Welcome to WebLogic Server Administration Scripting Shell


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

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

[Name : rms01app, Description : null, expiry Date : null]
PASSWORD:retail
*The above means for map rpm14 in APPDomain, alias DB-ALIAS points to database
user rms01app with a password of retail
```

updateCred

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

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

createCred

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

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

deleteCred

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

Script Mode Syntax

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

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

modifyBootStrapCredential

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

This script is available in interactive mode only.

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

Any output regarding the audit service can be disregarded.

addBootStrapCredential

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

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

This script is available in interactive mode only.

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

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

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

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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm14	<rpm batch user name> is the alias. Yes, here alias name = user name	<rpm batch user name>	App, batch use	Installer	RETAIL.USER	
	JAVA	<retail_home>/orpatch/config/javaapp_rpm							Each alias must be unique
			retail_installer	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	weblogic-alias	
			retail_installer	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			retail_installer	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			retail_installer	<LDAP-ALIAS>	cn=rpm.admin,cn=Users,dc=us,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALIAS	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name, ex: reim14>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name, ex: reim14>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name, ex: reim14>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name, ex: reim14>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			<installed app name, ex: reim14>	<LDAP-ALIAS>	cn=REIM.ADMIN,cn=Users,dc=us,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALIAS	
	JAVA	<retail_home>/orpatch/config/javaapp_reim							Each alias must be unique
			retail_installer	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			retail_installer	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			retail_installer	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			retail_installer	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			retail_installer	<LDAP-ALIAS>	cn=REIM.ADMIN,cn=Users,dc=us,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALIAS	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RESA app	DB credential store		Map=resa14 or what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file. The bootstrap directory under this directory has bootstrap cwallet.sso file.
RESA app	JAVA	<weblogic domain home>/retail/<deployed resa app name>/config							Each alias must be unique
			<installed app name>	<resa weblogic user alias>	<resa weblogic user name>	App use	Installer	wlsalias	
			<installed app name>	<resa schema db user alias>	<rmsdb shema user name>	App use	Installer	Resadb-alias	
			<installed app name>	<resa schema user alias>	<rmsdb shema user name>>	App use	Installer	resa-alias	
	JAVA	<retail_home>/orpatch/config/javaapp_resa							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			retail_installer	<resa weblogic user alias>	<resa weblogic user name>	App use	Installer	wlsalias	
			retail_installer	<resa schema db user alias>	<rmsdb shema user name>	App use	Installer	Resadb-alias	
	JAVA	<retail_ home>/orpatch/config/javaapp_rasrm							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
Alloc app	DB credential store		Map=alloc 14 or what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file. The bootstrap directory under this directory has bootstrap cwallet.sso file.
Alloc app	JAVA	<weblogic domain home>/retail/config							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms schema user alias>	<rms schema user name>	App use	Installer	dsallocAlias	
			<installed app name>	<alloc batch user alias>	<SYSTEM_ ADMINISTRATOR>	Batch use	Installer	alloc14	
	JAVA	<retail_ home>/orpatch/config/javaapp_alloc							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			retail_installer	<rms schema user alias>	<rms schema user name>	App use	Installer	dsallocAlias	
			retail_installer	<alloc batch user alias>	<SYSTEM_ ADMINISTRATOR>	Batch use	Installer	alloc14	
	JAVA	<retail_ home>/orpatch/config/javaapp_rasrm							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
SIM app	DB credential store		Map=oracle.retail.sim	Aliases required for SIM app use					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file.
	JAVA	<weblogic domain home>/retail/<deployed sim app name>/batch/resources/conf	oracle.retail.sim	<sim batch user alias>	<sim batch user name>	App use	Installer	BATCH-ALIAS	
	JAVA	<weblogic domain home>/retail/<deployed sim app name>/wireless/resources/conf	oracle.retail.sim	<sim wireless user alias>	<sim wireless user name>	App use	Installer	WIRELESS-ALIAS	
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	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
WebLogic			rib-<app>-app-server-instance	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integration use	Installer	weblogic-alias	
Admin GUI			rib-<app>#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integration use	Installer	admin-gui-alias	
Application			rib-<app>#user-alias	<app weblogic user alias>	<app weblogic user name>	Integration use	Installer	app-user-alias	Valid only for aip, rpm, sim
DB			rib-<app>#app-db-user-alias	<rib-app database schema user alias>	<rib-app database schema user name>	Integration use	Installer	db-user-alias	Valid only for rfm, rms, rwms, tafr
Error Hospital			rib-<app>#hosp-user-alias	<rib-app error hospital database schema user alias>	<rib-app error hospital database schema user name>	Integration use	Installer	hosp-user-alias	
RFI	Java	<RFI-HOME>/retail-financial-integration-solution/service-based-integration/conf/security							
			<installed app name>	rfiAppServerAdminServerUserAlias	<rfi weblogic user name>	App use	Installer	rfiAppServerAdminServerUserAlias	
			<installed app name>	rfiAdminUiUserAlias	<ORFI admin user>	App use	Installer	rfiAdminUiUserAlias	
			<installed app name>	rfiDataSourceUserAlias	<ORFI schema user name>	App use	Installer	rfiDataSourceUserAlias	

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

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

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

Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM)
2. Oracle Retail Sales Audit (ReSA)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Invoice Matching (ReIM)
7. Oracle Retail Price Management (RPM)

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

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

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

13. Oracle Retail Predictive Application Server (RPAS)
14. Oracle Retail Demand Forecasting (RDF)
15. Oracle Retail Category Management (RCM)
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 Analytics
26. Oracle Retail Advanced Science Engine (ORASE)
27. Oracle Retail Integration Bus (RIB)
28. Oracle Retail Service Backbone (RSB)
29. Oracle Retail Financial Integration (ORFI)
30. Oracle Retail Point-of-Service (ORPOS)
    - Oracle Retail Mobile Point-of-Service (ORMPOS) (requires ORPOS)
31. Oracle Retail Markdown Optimization (MDO)
32. Oracle Retail Clearance Optimization Engine (COE)
33. Oracle Retail Analytic Parameter Calculator for Markdown Optimization (APC-MDO)
34. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
35. Oracle Retail Macro Space Planning (MSP)

    The Oracle Retail Enterprise suite includes Macro Space Planning. This can be installed independently of and does not affect the installation order of the other applications in the suite. If Macro Space Planning is installed, the installation order for its component parts is:
    - Oracle Retail Macro Space Management (MSM)
    - Oracle Retail In-Store Space Collaboration (ISSC) (requires MSM)
    - Oracle Retail Mobile In-Store Space Collaboration (requires MSM and ISSC)