Oracle® Retail Active Retail Intelligence

Installation Guide

Release 13.2

E22172-04

June 2014

Oracle® Active Retail Intelligence Installation Guide, Release 13.2

Primary Author: Wade Schwarz

Contributors: Nathan Young

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

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

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

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

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

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

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

Contents

Send Us Your Comments

Oracle Retail Active Retail Intelligence Installation Guide, Release 13.2

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

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

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

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

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the 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 13.2 documentation set:

- *Oracle Retail Active Retail Intelligence Online Help*
- *Oracle Retail Active Retail Intelligence Operations Guide*
- *Oracle Retail Active Retail Intelligence Release Notes*
- *Oracle Retail Active Retail Intelligence User Guide*

See also:

- *Oracle Retail Merchandising Implementation Guide*

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.2) or a later patch release (for example, 13.2.1). If you are installing the base release and additional patch and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

Conventions

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

```
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 11gR2 Enterprise Edition. Options are: ▪ Oracle Enterprise Linux 5 Update 5 (OEL5.5) for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.5) for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.2) with the following specifications: **Components:** ▪ Oracle Partitioning ▪ Examples CD (Formerly the companion CD) **One off Patches:** ▪ 10170431 - CTWR CONSUMING LOTS OF CPU CYCLES Apply the following patch to RDBMS home if ASM is used. ▪ 11808931 - MERGE REQUEST ON TOP OF 11.2.0.2.0 FOR BUGS 10410054 10422126 **Other components:** ▪ Perl compiler 5.0 or later ▪ X-Windows interface

Check Supported Application Server Requirements

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

Supported on:	Versions Supported:
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.4). Options are: ▪ Oracle Linux 5 Update 3 (OL5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or Logical Domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.4) with the following one off patches. ▪ New Opatch version for Linux 64-bit. ▪ Patch 12361041: FRM-93652 while calling the scroll bar property from FORMS 11.1.1.4. **Note:** Apply any prerequisite patches mentioned in the README for Patch 12361041. ▪ Patches specific to OS: – Patch 13403345 for Sparc Solaris – Patch 13032871 for HP-UX Itanium – Patch 12836110 AIX **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.3) ▪ Oracle Forms Services 11g Release 1 (11.1.1.3) ▪ Java: JDK 1.6.0+ 64 bit or ▪ Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only. **IMPORTANT:** If there is an existing WebLogic 10.3.3 installation on the server, you must upgrade WebLogic 10.3.3 to WebLogic 10.3.4. All middleware components associated with WebLogic server 10.3.3 should be upgraded to 11.1.1.4. Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed. For information about how to complete the upgrade to WebLogic 10.3.4, see the My Oracle Support document, "How to Upgrade from WebLogic11g 10.3.3 to WebLogic11g 10.3.4" (ID 1432575.1). **Optional (SSO required)** ▪ Oracle SSO Server 10.1.4.3 ▪ Oracle Internet Directory 10g (10.1.4.3) or Oracle Internet Directory 11g (11.1.1.5) (must have separate Weblogic 10.3.5 for

Supported on:	Versions Supported:
	OID 11g) **Other components:** ▪ Oracle BI Publisher 10g (10.1.3.4)

Verify Single Sign-On

If Single Sign-On is to be used, verify the Oracle Infrastructure Server 10g has been installed. Verify the Oracle WebTier 11g server hosting Oracle Forms is registered with the Infrastructure Oracle Internet Directory.

Check Supported Web Browser and Client Requirements

General requirements for client running ARI include:

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

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)/Oracle Retail Trade Management (RTM)/Oracle Retail Sales Audit (ReSA)	13.2

Create a UNIX User Account to Install the Software

To create a UNIX User account, do the following.

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

To create the staging directory for ARI database files, do the following.

1. Create a staging directory for the ARI database installation software. There should be a minimum of 510 MB disk space available.
2. Copy the ari132dbserver.zip file from the CD/dbserverunix directory to the staging directory. This will be referred to as INSTALL_DIR for database installation tasks.
3. Change directories to INSTALL_DIR and extract the ari132dbserver.zip file.

Create Staging Directory for ARI Application Files

To create the staging directory for ARI application files, do the following.

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 ari132appserver.zip from the CD/appserverunix directory to the staging directory. This is referred to as INSTALL_DIR for application installation tasks.
3. Change directories to INSTALL_DIR and extract the file ari132appserver.zip.
4. Confirm that all scripts in INSTALL_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 WLS and Database installations
- Real Application Cluster Database and WebLogic Server Clustering

The Oracle Retail products have been validated against an 11.2.0.1 RAC database. When using a RAC database, all JDBC connections should be configured to use OCI connections rather than THIN connections. It is suggested that when using OCI connections, the Oracle Retail products database be configured in the tnsnames.ora file used by the WebLogic Server installations.

Clustering for WebLogic Server 10.3.3 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 11.2.0.1 Oracle Internet Directory database with the WebLogic 10.3.3 cluster. It is suggested that a Web Tier 11.1.1.3 installation be configured to reflect all application server installations if SSO will be utilized. References for Configuration:

- Oracle® Fusion Middleware High Availability Guide 11g Release 1 (11.1.1) Part Number E10106-09
- Oracle® Real Application Clusters Administration and Deployment Guide 11g Release 2 (11.2) Part Number E16795-08

3

Database Installation Tasks

It is assumed that Oracle 11g release 2, with appropriate patches, has already been installed. If not, refer to the section, “Check Supported Database Server Requirements,” in Chapter 1, “Preinstallation Tasks,” before proceeding. Also, INSTALL_DIR in this section refers to the directory created in the section, “Create Staging Directory for ARI Database Files,” in Chapter 1.

The following instructions assume that ARI will be installed in an existing RMS database.

> **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.

Verify the Existence of Oracle Packages

Confirm that the DBMS_SESSION, DBMS_RANDOM, DBMS_ALERT, DBMS_PIPE, DBMS_JOB, and UTL_SMTP 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','UTL_SMTP');
```

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. Refer to Appendix B.

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

Create ARI Schemas

To create ARI schemas, do the following.

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 'ARI132M'. This is referred to as the <master schema owner>.
2. Change directories to INSTALL_DIR/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 'ARI132G'. This will be referred to as the <generated schema owner>.
5. Change directories to INSTALL_DIR/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=pkols05
NLS_LANG	The locale setting for Oracle database client. Example: NLS_LANG=AMERICAN_AMERICA.UTF8
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.x/WLS/asinst_1

Create ARI Database Objects

To create ARI database objects, do the following.

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 INSTALL_DIR/ddl.
2. Log into SQL*Plus as <master schema owner> and execute ari13.sql. Review ari13.log for errors and correct as needed.
3. Change directories to INSTALL_DIR/db_objects.
4. Log into SQL*Plus as <master schema owner> and execute ari13dbo.sql. Review ari13dbo.log for errors and correct as needed.
5. Log into SQL*Plus <master schema owner> as and execute INSTALL_DIR/utility/inv_obj_comp.sql to validate any objects that may have become invalid. As mentioned above, some objects will still remain invalid until the dependent objects are created later.

Create ARI Data

To create ARI data, do the following.

1. Change directories to INSTALL_DIR/data.
2. Log into SQL*Plus as <master schema owner> and execute ari13ctl.sql. This script calls several scripts one of which is ari_options.sql. When prompted, enter values for the master and generated schema names when indicated.
3. Upon completion, check the spool file, ari13ctl.log, to verify that no errors were received.
4. Change directories to INSTALL_DIR/ctl.
5. Log into SQL*Plus as <master schema owner> and execute ari13scripts.sql.
6. Upon completion, check the spool file, ari13scripts.log, to verify that no errors were received.
7. Change directories to INSTALL_DIR/data/form_menu_elements.
8. Log into SQL*Plus as ARI 13 MASTER and run the following command:

```
SQL> @base_form_menu_elements.sql
```

Insert Language Data

Instructions below include how to insert secondary language data and primary language data.

Secondary Language Data

To insert secondary language data, do the following.

> **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 INSTALL_DIR/data/forms_menu_elements
2. Log in to sqlplus as ARI 13 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

To insert primary language data, do the following.

> **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 INSTALL_DIR/data/data_<lang>
2. Set the SQL*Plus session so that the encoding component of the NLS_LANG is UTF8.

   Example: RUSSIAN_RUSSIA.UTF8
3. Log into SQL*Plus as ARI 13 MASTER and run the following command:

   ```
   SQL> @ ari13_[lang].sql
   ```
4. Check the log file ari13_[lang].log for any errors.
5. Change directories to INSTALL_DIR/data/forms_menu_elements
6. Log into SQL*Plus as ARI 13 MASTER and run the following command:

   ```
   SQL> @base_form_menu_elements_langs_[lang].sql
   ```

Alter ARI Data

To alter ARI data, do the following.

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

   SQL> @mts_realm.sql

   SQL> @mts_parm_type.sql

   SQL> @mts_parm.sql

   SQL> @update_group_lookup.sql

Create Generated Schema Synonyms

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

1. Change directories to INSTALL_DIR/utility.
2. Log into SQL*Plus as < generated schema owner> and execute 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 INSTALL_DIR/utility.
2. Log into SQL*Plus as sys and run the following script:

   ```
   SQL> @revoke_install_privs.sql
   ```

Create User Synonyms

Each user of ARI 13.2 requires synonyms to the [MASTER] schema objects and the product(s) that it will be integrated with (e.g. RMS). After ensuring that each user has the 'create synonym' system privilege, create synonyms to each [MASTER] schema object of type table, view, function, package, procedure or sequence. This script prompts for values for the master schema names, the user name, password and database of the user you are creating the synonyms for. The user_syns.sql will create the synonyms to the [MASTER] schema objects. Create the additional synonyms (e.g. to RMS schema) as well.

1. Change directories to INSTALL_DIR/utility.
2. Log into SQL*Plus as <ARI Master Schema Owner> and run the following script:
   ```
   SQL> @user_syns.sql
   ```

4

Application Installation Tasks

These instructions assume that WebLogic 10.3.4 has been already been installed. If not, refer to “Check Supported Application Server Requirements” in Chapter 1, “Preinstallation Tasks” before proceeding. Also, INSTALL_DIR in this section refers to the directory created in the section, “Create Staging Directory for ARI Application Files,” in Chapter 1.

> **IMPORTANT**: If there is an existing WebLogic 10.3.3 installation on the server, you must upgrade WebLogic 10.3.3 to WebLogic 10.3.4. All middleware components associated with WebLogic server 10.3.3 should be upgraded to 11.1.1.4.
>
> Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.
>
> For information about how to complete the upgrade to WebLogic 10.3.4, see the My Oracle Support document, "How to Upgrade from WebLogic11g 10.3.3 to WebLogic11g 10.3.4" (ID 1432575.1).

Set Environment Variables

To set environment variables, do the following.

> **Note:** ORACLE_HOME is the location where WebLogic has been installed
> **Note:** ORACLE_INSTANCE is the location where WebLogic has been installed and contains the executables to compile forms

1. The T2kMotif.rgb file that is sent out with WebLogic (10.3.4) must be modified. It located at the following location:

   $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin

   Make a copy of the file Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).

   Modify the file Tk2Motif.rgb file so that it contains the following line:

   ```
   Tk2Motif*fontMapCs: iso8859-2=UTF8
   ```
2. Log on to the application server as the oretail user.
3. Set the DISPLAY variable to the IP address plus “:0.0” (for example, 10.1.1.1:0.0) of the application server.
4. Set the following variables:

> **Note:** ORACLE_HOME is the location where Oracle WebLogic (10.3.3) has been installed.

> **Note:** ORACLE_INSTANCE is the location where WebLogic has been installed and contains the executables to compile forms

- All OS Platforms
  - PATH=$ORACLE_HOME/bin:$DOMAIN_HOME/opmn/bin:$DOMAIN_HOME/dcm/bin:INSTALL_DIR/forms11g_scripts:$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 schema owner>/<ARI schema password>@<ARI database>

    Or using the wallet UP=/@dvols29_ari01user

> **Note:** Verify that TNS is set up correctly by using the UP variable to successfully log into the ARI 13 schema.

> **Example:** /u00/oracle> sqlplus $UP

Install ARI Forms

To install ARI Forms, do the following.

1. Copy all libraries (.pll files) in the INSTALL_DIR/forms/src directory to the directories to the INSTALL_DIR/forms/bin directory.
2. Change directories to INSTALL_DIR/forms/bin.
3. 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

4. Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully).
5. Remove all newly created .plx files.
6. Copy all forms (*.fmb files) in the INSTALL_DIR/forms/src directory to the INSTALL_DIR/forms/bin directory.

7. Run fmb2fmx11g_fm (in INSTALL_DIR/forms/bin) to compile the ARI reference forms.

   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

   > **Note:** Warnings similar to the following can be ignored:
   >
   > FRM-30029: Cannot find window W_REFERENCE.
   > View Canvas C_TOOLBAR
   > Form: FM_EDIT
   >
   > FRM-30085: Unable to adjust form for output.
   > Form not created

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

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

9. Remove all newly created fm_*.fmx files (reference forms should not have executable files).

10. Run fmb2fmx11g (in INSTALL_DIR/forms/bin) to generate ARI runtime forms – .fmx's.

    > **Note:** Warnings can be ignored if corresponding .fmx files are successfully created. For example:
    >
    > FRM-30351: No list elements defined for list item.
    > List REALM_TYPE_ID
    > FRM-30351: No list elements defined for list item.
    > List DATA_TYPE
    >
    > Created form file metadata.fmx

11. 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).

12. 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
    ```

13. Copy all menus (*.mmb files) in the INSTALL_DIR/forms/src directory to the INSTALL_DIR/forms/bin directory.
14. Run mmb2mmx11g (in INSTALL_DIR/forms/bin) to generate ARI runtime menus – .mmx's.
15. Check to make sure that each .mmb file has a corresponding .mmx file.
16. 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 10.3.4 for ARI

To configure WebLogic 10.3.4 for ARI, complete the following steps.

Note: The proper WebLogic 10.3.4 components must be started in order to run Forms applications.

Note: ORACLE_HOME refers to the location where WebLogic 10.3.4 is installed.

1. Make a copy of the file ORACLE_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/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 due to changes between older version of Oracle Forms and WebLogic 10.3.4.

Example:

```
FORMS_PATH=/u00/forms/bin:/u00/oracle/product/10.3.x_WLS/WLS/asinst_1/FormsComponent/forms
NLS_DATE_FORMAT=DD-MON-RR

NLS_LANG=AMERICAN_AMERICA.UTF8

FORMS_USERNAME_CASESENSITIVE=1
FORMS_REJECT_GO_DISABLED_ITEM=FALSE
```

3. Modify the file formsweb.cfg located at ORACLE_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/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,icons.jar
imageBase=codeBase

                    form=arimstr.fmx
```

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

- Set ssoMode to true.
- 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=true

ssoDynamicResourceCreate=true
```

To allow the icons jar to be picked up in your forms the following line needs to be added to formsweb.cfg as well.

# Forms applet archive setting for other clients (Sun Java Plugin, Appletviewer, etc)

archive=frmall.jar, icons.jar

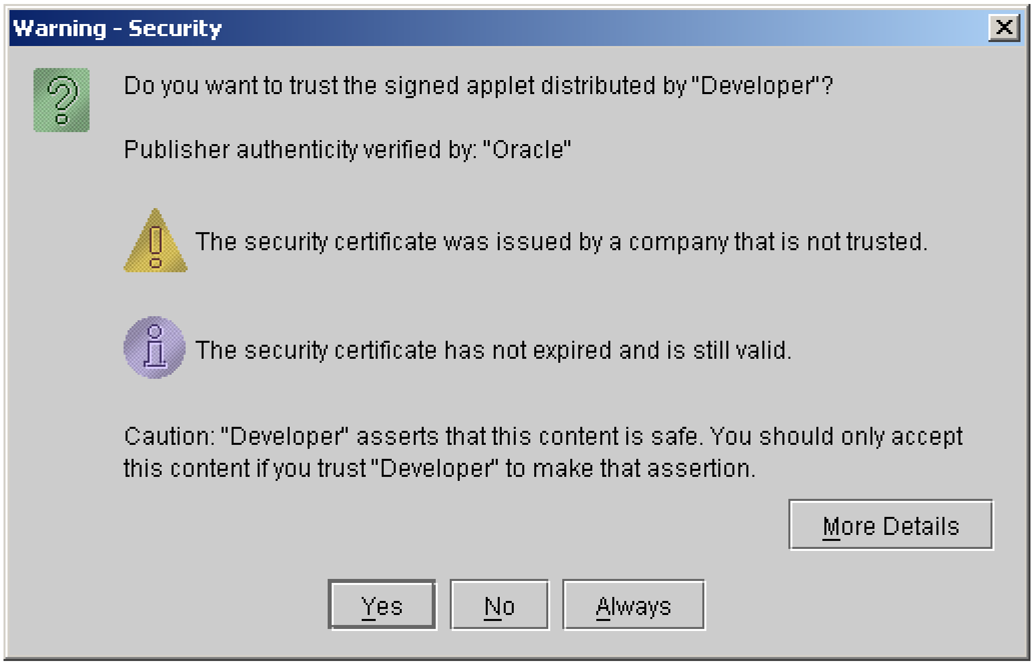
4. cd to /INSTALL_DIR/web_html/WebHelp and do the following:
   - Place the icons.jar file into directory $WEBLOGIC_HOME/as_1/forms/java
   - Indicate to Forms runtime where the icons are to be fetched: WEBLOGIC_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/ applications/formsapp_11.1.1/config/forms/registry/oracle/forms/registry

     default.icons.iconpath=
     default.icons.iconextension=gif
5. Launch ARI by entering the following URL in a browser. Prior to testing, the Sun JRE 1.6.0_22+ 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 1 (11.1.1.4) 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 2 above).

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

Note: The first time ARI is accessed, the user may be prompted with the following security warning. Click **Yes**.



Do you want to trust the signed applet distributed by "Developer"?

Publisher authenticity verified by: "Oracle"

The security certificate was issued by a company that is not trusted.

The security certificate has not expired and is still valid.

Caution: "Developer" asserts that this content is safe. You should only accept this content if you trust "Developer" to make that assertion.

6. 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 created in Chapter 2

Example: Username: ARI13M
Password: retek
Connect String: prod_db1

Install Helpfile

To install Helpfile, do the following.

1. Log in to the WebLogic Admin console to which online help will be installed.
2. Create a server. In this example **ari-help-server** is being used.
3. On the left in the console, select Environment/Machines and add the new managed server, ari-help-server
4. Click Control tab (after the server, ari-help-server, has been created) and check the server. Click Start.
5. Click Deployments on left followed by Lock and Edit and Install.
6. Select **Install this Deployment as an Application**.
7. Select the managed server, ari-help-server, that was created earlier.
8. Select the remaining defaults. Click **Finish**.
9. Select deployment. Click **Start**.

Log in to sqlplus as the ARI 13 master schema owner (ARI13M) 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:7778' where lang=1;

Install Oracle Configuration Manager

This Oracle Retail product has been instrumented for configuration discovery and collection by Oracle Configuration Manager (OCM). It is recommended that you take advantage of OCM in your environment. See the My Oracle Support document, "Oracle Configuration Manager Installer Guide" (ID 1071030.1), for more information about OCM instrumentation in Oracle Retail products.

To installing OCM for ARI, do the following.

1. Add this new ARI application to the retail inventory:

   Create or modify the $ORACLE_HOME/retail_inventory/oracle_retail_ARIApp.properties file. Add the following lines to this file, substituting where appropriate:

   <IAS_name>.<ARI_name>+ARI_INSTALL_DATE=<yyyy-MM-dd HH:mm:ss>

   <IAS_name>.<ARI_name>+ARI_VERSION=13.2.0

   > **Example:** OHWLS1033.mspdev69.ari13inst1+ARI_INSTALL_DATE=2007-11-10 09:51:50
   >
   > OHWLS1033.mspdev69.ari13inst1+ARI_VERSION=13.2.0

   <IAS_name> refers to the name of the WebLogic application server. This name was set when the application server software was installed.

   <yyyy-MM-dd HH:mm:ss> refers to the date of ARI installation. See example value above.

   <ARI_name> refers to a unique name to identify this ARI forms installation. This is arbitrary and can be chosen at the time that this oracle_retail_ARIApp.properties file is created.

   > **Note:** If there are multiple installations of ARI using the same application server ORACLE_HOME, then there may be multiple sets of properties oracle_retail_ARIApp.properties with different <ARI_name> values.

   The <IAS_name>.<ARI_name> string used in this file becomes the OCM *target name* for this ARI installation.

2. Determine whether or not the OCM collector has been installed in the application server ORACLE_HOME. Check for the existence of an $ORACLE_HOME/ccr directory containing the collector software. If there is already an OCM collector installed in this location then the rest of these steps can be skipped.
3. If there is not yet an OCM collector installed in the application server ORACLE_HOME, the Retail OCM Installer should be used to install it. Use the retail-OCM-withAnt.zip file included with the ARI release and follow the instructions in the *Oracle Configuration Manager Installer Guide* (Doc ID: 835024.1) to run the Retail OCM Installer independently.

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. Note that 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.

Test the ARI Application

Oracle Retail provides test cases that allow you to smoke test your installation. Refer to the My Oracle Support document, “Oracle Retail Merchandising Installation Test Cases” (ID 1277131.1).

A

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
```

B

Appendix: Oracle Single Sign-On

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

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

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

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

What Do I Need for Oracle Single Sign-On?

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

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle HTTP Server 11g Release 1 as a front end to the Oracle WebLogic Server. The Oracle HTTP Server is included in the Oracle Web Tier Utilities 11g Release 1 (11.1.1).
- An Oracle Single Sign-On Plug-in, used to authenticate the user and create the OSSO session cookie. This is available in the Oracle Fusion Middleware 11g Web Tier Utilities (11.1.1.20) package.
- The Delegated Administration Services (DAS) application, used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OSSO system and registering HTTP servers.

For more information on setting up single sign on, refer to the *Oracle Retail Predictive Application Server Installation Guide*.

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

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

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

Oracle Single Sign-on Terms and Definitions

The following terms and definitions apply to Oracle Single Sign-on.

Authentication

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

Dynamically Protected URLs

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

Identity Management Infrastructure

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

MOD_OSSO

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

MOD_WEBLOGIC

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

Oracle Internet Directory

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

Partner Application

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

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

Realm

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

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

Statically Protected URLs

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

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

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

What Single Sign-On is not

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

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

How Oracle Single Sign-On Works

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

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

Statically Protected URLs

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

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

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

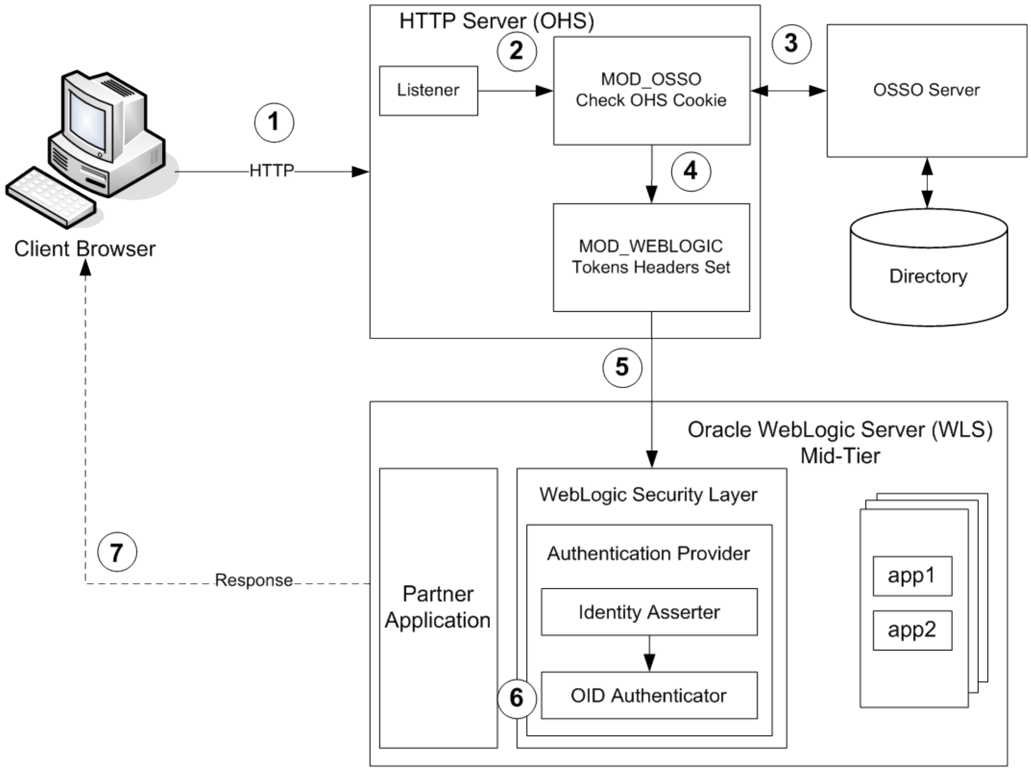
Dynamically Protected URLs

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

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

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

Single Sign-on Topology



Installation Overview

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

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

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

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

Infrastructure Installation and Configuration

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

OID User Data

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

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

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

OID with Multiple Realms

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

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

User Management

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

OID DAS

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

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

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

LDIF Scripts

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

User Data Synchronization

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

C

Appendix: 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.

D

Appendix: Setting Up Password Stores with Oracle Wallet

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

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

About Password Stores and Oracle Wallet

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

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

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

```
sqlplus /@db_username
```

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

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

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

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

Setting Up Password Stores for Database User Accounts

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

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

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

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

4. 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.*

5. 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.

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

7. 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
   ```

8. 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, RPM Plsql Batch, RETL DB, RWMS batch, and ARI
- For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For RMS, RPM Plsql Batch, RETL DB, RWMS batch, and ARI

To set up wallets for database accounts for RMS, RPM PlsqlBatch, RETL DB, RWMS batch, and ARI, do the following.

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

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

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

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

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

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

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

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

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

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

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

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

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

   Example: `mkstore –wrl . –createCredential dvols29_rms01user rms01user passwd`
6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.
   ```
   $ export TNS_ADMIN=/projects/rms13.2/dev/.wallet /* This is very import to use
   wallet to point at the alternate tnsnames.ora created in this example */

   $ sqlplus /@dvols29_rms01user

   SQL*Plus: Release 11

   Connected to:
   Oracle Database 11g

   SQL> show user
   USER is "rms01user"
   ```

   Running batch programs or shell scripts would be similar:
   ```
   Ex: dtesys /@dvols29_rms01user
   script.sh /@dvols29_rms01user

   Set the UP unix variable to help with some compiles :

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

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

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet
  ```
  mkstore –wrl . –deleteCredential dvols29_rms01user
  ```
- Change the password for a credential on wallet
  ```
  mkstore –wrl . –modifyCredential dvols29_rms01user rms01user passwd
  ```

- List the wallet credential entries

  ```
  mkstore –wrl . –list
  ```

  This command returns values such as the following.

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

- View the details of a wallet entry

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

  Returns the value of the entry:

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

  Returns value of the entry:

  ```
  rms01user
  ```

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

  Returns value of the entry:

  ```
  passwd
  ```

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

For Java 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: mspdv351:[103x_WLS] /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config
- Application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.
- Scripts are located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/retail-public-security-api/bin for administering wallet entries.

  Example:

  mspdv351:[103x_WLS] /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/retail-public-security-api/bin
- In this directory is a script to help you update each alias entry without having to remember the wallet details. For example, if you set the RPM database alias to rms01user, you will find a script called update-RMS01USER.sh.

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

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

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

  Example:

  mspdv351:[103x_WLS] /u00/webadmin/reim/application/retail-public-security-api/bin

update-<ALIAS>.sh

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

Usage:

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

Example:

```
mspdev71:[103xWLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/java_domain/retail/rpm1
32test/retail-public-security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass
in.
mspdev71:[103xWLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/java_domain/retail/rpm1
32test/retail-public-security-api/bin>
```

dump_credentials.sh

dump_credentials.sh is used to retrieve information from the 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 /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

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

Below are the credentials found in the wallet at the location:/u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

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

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

save_credential.sh

save_credential.sh is used to update the information in wallet. If you are unsure about the information that is currently in the wallet, use dump_credentials.sh as indicated above. You can add new or update using save_credential.sh as shown below:

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

Example:

```
mspdv351:[103x_WLS]
/u00/webadmin/mock132_testing/rtil/rtil/application/retail-public-security-
api/bin> save_credential.sh –l
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/reta
il/reim13/config
-a RMS-ALIAS -p reim13 -u rms132mock

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

Enter password:
Verify password:
```

Note: `-p` in the above command is for partition name. You must specify the proper partition name used in application code for each Java application.

save_credential.sh and dump_credentials.sh scripts are the same for all applications. If using save_credential.sh to add a wallet entry or to update a wallet entry, bounce the application/managed server so that your changes are visible to the application. Also, save a backup copy of your cwallet.sso file in a location outside of the deployment path, because redeployment or reinstallation of the application will wipe the wallet entries you made after installation of the application. To restore your wallet entries after a redeployment/reinstallation, copy the backed up cwallet.sso file over the cwallet.sso file. Then bounce the application/managed server.

Usage

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

How Does the Wallet Relate to the Application?

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

Reim.properties code sample:

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

csm.wallet.path=/u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa
_domain/retail/reim13/config
csm.wallet.partition.name=reim13
```

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 REIM app user reimbat, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments as needed by the program in question>.

Setting up RETL Wallets

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

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

To set up RETL wallets, perform the following steps:

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

- Re-enter the database password.
- Enter D to exit the setup script.

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

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

   - The RETL_WALLET_ALIAS should point to the Java wallet entry:

     ```
     export RETL_WALLET_ALIAS="retl_java_rms01user"
     ```

   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:

     ```
     export ORACLE_WALLET_ALIAS="dvols29_rms01user"
     ```

   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:

     ```
     export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"
     ```

5. To change a password later, run `setup-security-credential.sh`.

   - Enter 2 to update a database credential.
   - Select the credential to update.
   - Enter the database user to update or change.
   - Enter the password of the database user.
   - Re-enter the password.

Quick Guide for Retail Wallets

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

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
			<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
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip13	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip13	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip13	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm13	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rpm13	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
			rpm13	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbatch-alias	

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

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
			rms	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			rib-sim	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application
RETL	DB	<RETL home>/.wallet	n/a	<target application user alias>	<target application db userid>	App use	Manual	<db>_<user>	User may vary depending on RETL flow's target application
RIB	JAVA	<RIBHOME DIR>/deployment-home/conf/security							<app> is one of aip, rfm, rms, rpm, sim, rwms, tafr
JMS			jms<1-5>	<jms user alias> for jms<1-5>	<jms user name> for jms<1-5>	Integration use	Installer	jms-alias	
WebLogic			rib-<app>-app-server-instance	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integration use	Installer	weblogic-alias	
Admin GUI			rib-<app>#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integration use	Installer	admin-gui-alias	

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
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	

Appendix: Web Browser Configuration

This appendix provides information on configuring the Internet Explorer Web browser for operation with ARI.

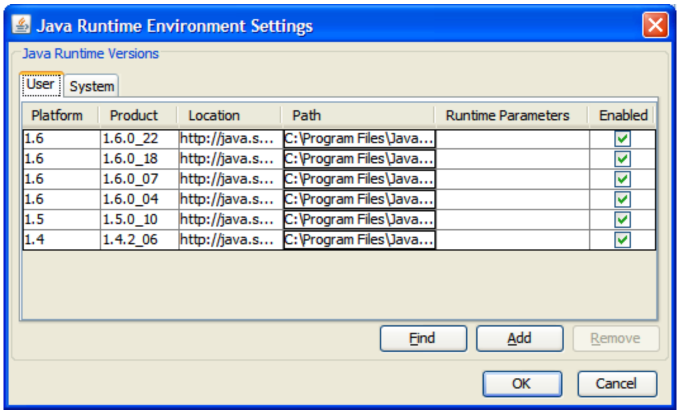
> **Note:** The Oracle's Java Runtime Environment (JRE) is required to support Applets within a sandboxed security environment in the Web browser. The security architecture of the JRE has changed from JRE1.6.0_18 and requires additional configuration in Windows.

Microsoft Internet Explorer Versions 7 and 8

> **Note:** Before proceeding, ensure that you have the JRE 1.6.0_22 installed.

To configure the latest JRE for Internet Explorer 7, do the following.

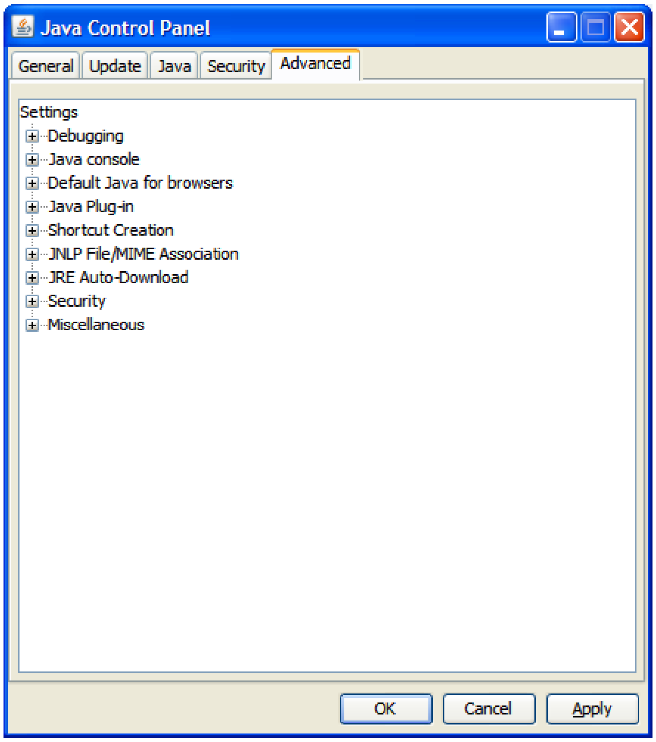
1. Use the Windows Control Panel and open the **Java** Control Panel.
2. Select the **Java** tab in the Java Control Panel, and click the **View** button to confirm that the JRE1.6.0_22 product is installed and enabled.



Platform	Product	Location	Path	Runtime Parameters	Enabled
1.6	1.6.0_22	http://java.s...	C:\Program Files\Java...		☑
1.6	1.6.0_18	http://java.s...	C:\Program Files\Java...		☑
1.6	1.6.0_07	http://java.s...	C:\Program Files\Java...		☑
1.6	1.6.0_04	http://java.s...	C:\Program Files\Java...		☑
1.5	1.5.0_10	http://java.s...	C:\Program Files\Java...		☑
1.4	1.4.2_06	http://java.s...	C:\Program Files\Java...		☑

Java Runtime Environment Settings Window

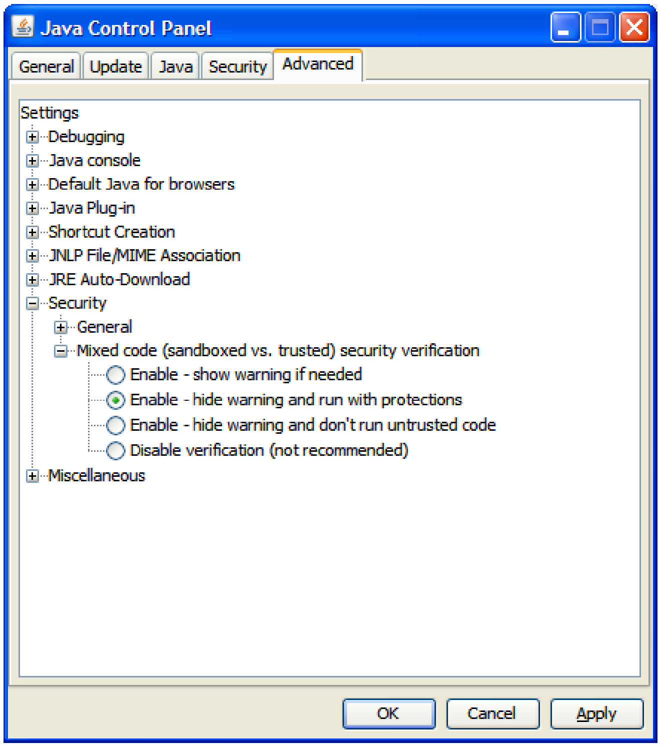
3. Click **OK** and return back to the Java Control Panel.
4. Click the **Advanced** tab.



Settings
- ⊞ Debugging
- ⊞ Java console
- ⊞ Default Java for browsers
- ⊞ Java Plug-in
- ⊞ Shortcut Creation
- ⊞ JNLP File/MIME Association
- ⊞ JRE Auto-Download
- ⊞ Security
- ⊞ Miscellaneous

Options in the Advanced Tab

5. Click the Plus (+) icon and expand **Security**, and then expand **Mixed code**.

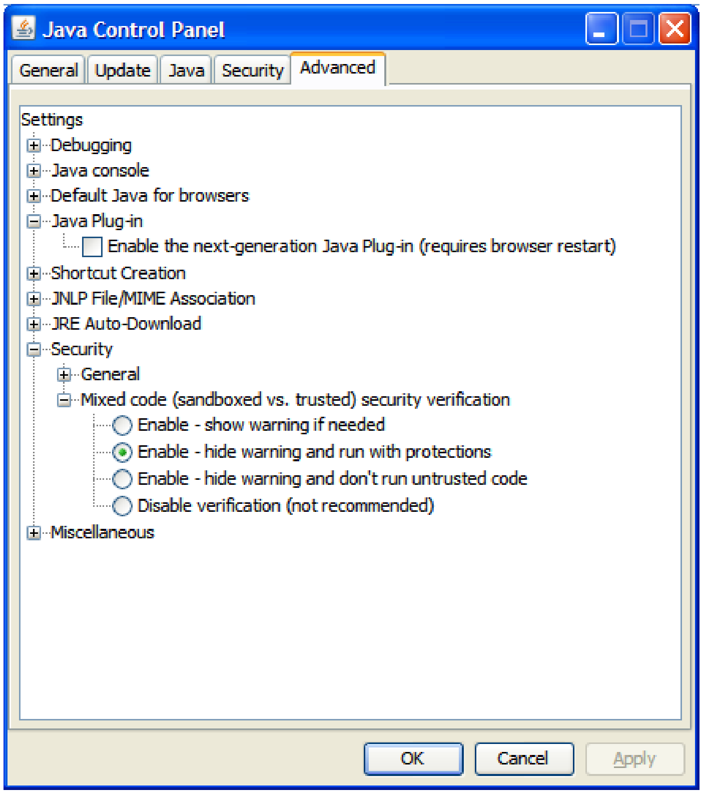


Settings
- ⊞ Debugging
- ⊞ Java console
- ⊞ Default Java for browsers
- ⊞ Java Plug-in
- ⊞ Shortcut Creation
- ⊞ JNLP File/MIME Association
- ⊞ JRE Auto-Download
- ⊟ Security
  - ⊞ General
  - ⊟ Mixed code (sandboxed vs. trusted) security verification
    - ○ Enable - show warning if needed
    - ● Enable - hide warning and run with protections
    - ○ Enable - hide warning and don't run untrusted code
    - ○ Disable verification (not recommended)
- ⊞ Miscellaneous

Mixed Code Settings in Java Control Panel

6. Click the **Enable – hide warning and run with protections** option to stop warning popup messages from appearing, but still enables the Java applet code to execute with protection.

7. Disable the next generation Java Plug-in option by expanding **Java Plug-in**, and then clearing the **Enable the next-generation Java Plug-in** check box.



Settings
- Debugging
- Java console
- Default Java for browsers
- Java Plug-in
  - □ Enable the next-generation Java Plug-in (requires browser restart)
- Shortcut Creation
- JNLP File/MIME Association
- JRE Auto-Download
- Security
  - General
  - Mixed code (sandboxed vs. trusted) security verification
    - ○ Enable - show warning if needed
    - ● Enable - hide warning and run with protections
    - ○ Enable - hide warning and don't run untrusted code
    - ○ Disable verification (not recommended)
- Miscellaneous

Java Plug-in Option in Java Control Panel

Restart your Web browser to run the RWMS Forms displays.

F

Appendix: Installation Order

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

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

Enterprise Installation Order

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

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

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

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

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

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

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

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