

Oracle® Retail Warehouse Management System
Installation Guide
Release 15.0.2
E90686-01

December 2017

Copyright © 2017, Oracle. All rights reserved.

Primary Author: Nathan Young

Contributors: Amandeep Bhatti, Santhosh Kumar NC

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

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	xi
Preface	xiii
Audience	xiii
Customer Support.....	xiii
Review Patch Documentation	xiii
Improved Process for Oracle Retail Documentation Corrections	xiii
Oracle Retail Documentation on the Oracle Technology Network.....	xiv
Conventions.....	xiv
1 Preinstallation Tasks	1
Requesting Infrastructure Software.....	1
Installation Terminology.....	1
Check Supported Database Server Requirements	2
Check Supported Application Server Requirements	3
Check Supported Web Browser and Client Requirements	4
Supported Mobile Client Device Requirements	4
Supported Radio Frequency Device Requirements	5
Supported Oracle Retail Products	5
Supported Oracle Retail Integration Technologies	5
UNIX User Account Privileges to Install the Software	6
2 RAC and Clustering	7
3 Database Installation Tasks – Full	9
Create Staging Directory for RWMS Installer	9
Create the RWMS Database.....	9
Create the Database Instance Using the Oracle Generic Template.....	10
Create the Required RWMS Tablespaces	11
Create the Schema Owner for RWMS.....	12
Run the RWMS Database Schema Installation	13
4 Database Installation Tasks – Patch	15
Create Staging Directory for RWMS Installer	15
(Optional) Analyze Changes in the Patch.....	15
Run the RWMS Database Schema Installation	15
5 Application Server Installation Tasks – RWMS Forms	17
Prepare Application Server for RWMS	17
Create and Configure tnsnames.ora file	18
Configure Tk2Motif.rgb	18
Create Staging Directory for RWMS Installer	18
Install RWMS Oracle Forms	18
6 Application Server Installation Tasks – Patch	21
Create Staging Directory for RWMS Installer	21

(Optional) Analyze Changes in the Patch.....	21
Run the RWMS Application Installation	21
Install RWMS Oracle Forms	21
7 Enhanced Navigation and Dashboards Installation	25
Steps to Create the New Domain with ADF Libraries.....	25
Fusion Middleware Infrastructure and WebLogic Server 12c (12.2.1.2) Installation	25
Install RCU Database Schemas	36
Create a New ADF Domain (with managed server and EM).....	46
Start the Node Manager	62
Start the AdminServer (admin console).....	62
Start the Managed Server.....	63
Run the RWMS Installer.....	63
Custom Authentication Provider set up	65
Disable User Lock Out Settings.....	70
Application: Logging File Settings	70
8 Manual Installation and Configuration Tasks	73
Prerequisites	73
Path References Used in this Section.....	73
BI Server Component Installation Tasks.....	74
Installation Process Overview.....	75
BiPublisher 12c only - Installation Process Overview	75
Post install steps for BiPublisher 12C.....	75
Installing the RWMS BI Publisher Templates.....	79
Configuring the RWMS JDBC connection.....	81
Configuring Forms to use BI Publisher.....	84
Check the Configuration of formsweb.cfg	84
Configuring the Forms Installation Environment Files.....	86
Update <INSTALLATION_NAME>.env File With FORMS_DATETIME_LOCAL_TZ	91
RWMS Wallet Configurations.....	91
Application (Java) Wallet.....	92
Oracle Wallet	93
WLS Database Credential Store	93
Configuring Terminal Server for RWMS Application	93
Setup and Configure JNLP Launch Shortcut for Kiosk Mode.....	93
Configure Java Security Exceptions and Certificate Imports	94
Configuring Terminal Server Close for Radio Frequency Devices	94
File Placement	95
Update the RF Launch Modules	95
rwms_rf_menu.html Module	95
close.html Module	96
Update fmrweb.res for Keymapping	98

9	Web Services Installation	99
	Create a Managed Server.....	99
	Create a Datasource.....	99
	Deploy RWMS Service EAR File.....	99
	Configure Web Service Security	100
10	Patching Procedures	103
	Oracle Retail Patching Process	103
	Supported Products and Technologies	103
	Patch Concepts	104
	Patching Utility Overview	105
	Changes with 15.0.....	105
	Patching Considerations	105
	Patch Types.....	105
	Incremental Patch Structure	106
	Version Tracking.....	106
	Apply all Patches with Installer or ORPatch.....	106
	Environment Configuration	107
	Retained Installation Files.....	107
	Reloading Content.....	107
	Java Hotfixes and Cumulative Patches	107
	Backups	108
	Disk Space.....	108
	Patching Operations	109
	Running ORPatch	109
	Merging Patches.....	119
	Compiling Application Components.....	120
	Deploying Application Components	122
	Maintenance Considerations	123
	Database Password Changes.....	123
	WebLogic Password Changes.....	124
	Infrastructure Directory Changes.....	125
	DBManifest Table.....	125
	RETAIL_HOME relationship to Database and Application Server.....	125
	Jar Signing Configuration Maintenance	125
	Customization	127
	Patching Considerations with Customized Files and Objects	127
	Registering Customized Files.....	128
	Custom Compiled Java Code.....	130
	Extending Oracle Retail Patch Assistant with Custom Hooks	132
	Troubleshooting Patching.....	137
	ORPatch Log Files.....	137
	Restarting ORPatch.....	137

Manual DBManifest Updates.....	137
Manual Restart State File Updates	139
DISPLAY Settings When Compiling Forms.....	139
JAVA_HOME Setting	139
Patching Prior to First Install.....	139
Providing Metadata to Oracle Support.....	140
A Appendix: Oracle Database 12cR1 Parameter File	143
B Appendix: Tablespace Creation Scripts.....	145
Non-Encrypted Tablespace Creation	145
Encrypted Tablespace Creation	145
Configure a Wallet:.....	146
Encryption at Tablespace Level	147
C Appendix: Sample Oracle Net Files for the Server.....	149
listener.ora.....	149
tnsnames.ora	150
D Appendix: RWMS Database Schema Installation Screens.....	151
E Appendix: RWMS Application (forms) Installation Screens	161
F Appendix: RWMS Enhanced Navigation and Dashboards Installation Screens.....	173
G Appendix: RWMS Analyze Tool.....	193
Run the RWMS Analyze Tool	193
H Appendix: Installer Silent Mode	195
I Appendix: Common Installation Errors.....	197
Database Installer Hangs on Startup.....	197
Warning: Could not create system preferences directory	197
WLS_FORMS server failed to restart	198
Warning: Couldn't find X Input Context.....	198
FRM-30085: Unable to adjust form for output	199
Unresponsive Drop-Downs.....	200
Could not execl robot child process: Permission denied.....	201
ConcurrentModificationException in Installer GUI.....	201
ORA-04031 (unable to allocate memory) error during database schema installation	201
X Error of failed request: BadWindow (invalid Window parameter)	202
RIB Errors.....	202
Error Connecting to Database URL	203
Cannot Run Reports in RWMS Application on HP Host.....	203
Frequent Session Timeouts in RWMS Application on HP Host.....	203
Screen Navigation Delay Using Java 7U85+ Between ADF and Forms Screens.....	204
J Appendix: URL Reference	207
JDBC URL for a Database	207
JNDI Provider URL for an Application	207

K	Appendix: Single Sign-On for WebLogic	209
	What Do I Need for Single Sign-On?	209
	Can Oracle Access Manager Work with Other SSO Implementations?	209
	Oracle Single Sign-on Terms and Definitions	210
	What Single Sign-On is not.....	211
	How Oracle Single Sign-On Works	211
	Installation Overview	213
	User Management.....	213
L	Appendix: Setting Up an Oracle Wallet.....	215
	Steps to Set up a pl/sql Wallet.....	215
	Additional Wallet Commands	217
	Setting up Application (Java) Wallet for RWMS .env File	217
	Setting Up WLS Database Credential Store	219
M	Appendix: Setting Up Password Stores with wallets/credential stores.....	225
	About Database Password Stores and Oracle Wallet	225
	Setting Up Password Stores for Database User Accounts.....	226
	Setting up Wallets for Database User Accounts	227
	For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI.....	227
	Setting up RETL Wallets	229
	For Java Applications (SIM, ReIM, RPM, RIB, AIP, Alloc, ReSA, RETL).....	230
	How does the Wallet Relate to the Application?	233
	How does the Wallet Relate to Java Batch Program use?.....	233
	Database Credential Store Administration.....	233
	Managing Credentials with WSLT/OPSS Scripts	235
	listCred	236
	updateCred	237
	createCred	238
	deleteCred.....	238
	modifyBootStrapCredential	238
	addBootStrapCredential	240
	Quick Guide for Retail Password Stores (db wallet, java wallet, DB credential stores)	241
N	Appendix: Manual Forms Compilation	251
O	Appendix: Installation Order	253
	Enterprise Installation Order.....	253

Send Us Your Comments

Oracle Retail Warehouse Management System, Installation Guide, Release 15.0.2

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

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

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

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

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

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com
Please give your name, address, electronic mail address, and telephone number (optional).

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

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

Preface

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

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, 15.0) or a later patch release (for example, 15.0.2). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

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

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

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

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part

number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

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

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

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

Preinstallation Tasks

This chapter includes steps to complete before installation.

Note: Oracle Retail assumes that the retailer has applied all required fixes for supported compatible technologies.

Requesting Infrastructure Software

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

Installation Terminology

STAGING_DIR – It is the directory where the `rwms15installer.zip` is copied and extracted locally.

RETAIL_HOME – It is the directory where Database Files are stored, and Forms are installed. This will contain the `ORPatch` directory as well.

- Database `RETAIL_HOME` - The location where RWMS Database Files are stored. This location will be used during the subsequent patching of the RWMS.
- Forms `RETAIL_HOME` - This is the Forms installation directory, the location where RWMS Forms are installed.

Note: A separate `RETAIL_HOME` is required for DB and Forms installations.

Check Supported Database Server Requirements

General Requirements for a database server running RWMS include the following.

Supported on	Versions Supported
Database Server OS	<p>OS certified with Oracle Database 12cR1 Enterprise Edition. Options are:</p> <ul style="list-style-type: none"> ▪ Oracle Linux 6 and 7 for x86-64 (actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 and 7 for x86-64 (actual hardware or Oracle virtual machine). ▪ AIX 7.1 (actual hardware or LPARs) ▪ Solaris 11.3 Sparc (actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Database Server 12C	<p>Oracle Database Enterprise Edition 12cR1 (12.1.0.2) with the following specifications:</p> <p>Components:</p> <ul style="list-style-type: none"> ▪ Oracle Partitioning ▪ Examples CD <p>Oneoffs:</p> <ul style="list-style-type: none"> ▪ 20846438: ORA-600 [KKPAPXFORMFKK2KEY_1] WITH LIST PARTITION ▪ 19623450: MISSING JAVA CLASSES AFTER UPGRADE TO JDK 7 ▪ 20406840: PROC 12.1.0.2 THROWS ORA-600 [17998] WHEN PRECOMPILING BY 'OTHER' USER ▪ 20925154: ORA-39126: WORKER UNEXPECTED FATAL ERROR IN KUPW\$WORKER GATHER_PARSE_ITEMS JAVA <p>RAC only:</p> <ul style="list-style-type: none"> ▪ 21260431: APPSST 12C : GETTING ORA-4031 AFTER 12C UPGRADE ▪ 21373473: INSTANCE TERMINATED AS LMD0 AND LMD2 HUNG FOR MORE THAN 70 SECS <p>Other components:</p> <ul style="list-style-type: none"> ▪ Perl interpreter 5.0 or later ▪ X-Windows interface ▪ JDK1.7

Note: By default, JDK is at 1.6. After installing the 12.1.0.2 binary, apply patch 19623450. Follow the instructions on Oracle Database Java Developer's Guide 12c Release 1 to upgrade JDK to 1.7. The Guide is available at: <http://docs.oracle.com/database/121/JJDEV/chone.htm#JJDEV01000>.

Check Supported Application Server Requirements

General requirements for an application server capable of running RWMS include the following.

Supported on	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 12c 12.2.1.2 Options are: <ul style="list-style-type: none"> ▪ Oracle Linux 6 and 7 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 and 7 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11.3 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Application Server	For Oracle Forms Based Screens Oracle Fusion Middleware 12c 12.2.1.2 Components: <ul style="list-style-type: none"> ▪ Oracle WebLogic Server 12c (12.2.1.2) (Installed as part of FMW Infrastructure 12c) ▪ Oracle Forms Services 12c (12.2.1.2) with required patches: <ul style="list-style-type: none"> ▪ Patch 25757471: SECURE EXTERNAL PASSWORD STORE (SEPS) DOES NOT WORK IN FORMS 12.2.1.2 ▪ Oracle Enterprise Manager Fusion Middleware Control 12c For Enhanced Navigation and Dashboards Application Components: <ul style="list-style-type: none"> ▪ Oracle WebLogic Server 12c (12.2.1.2) ▪ Oracle ADF (Application Development Framework) 12c (12.2.1.2) ▪ Oracle Enterprise Manager Fusion Middleware Control 12c Java: <ul style="list-style-type: none"> ▪ JDK 1.8+ 64 bit Optional (if SSO is required) <ul style="list-style-type: none"> ▪ Oracle WebTier 12c (12.2.1.2) ▪ Oracle Access Manager 11g Release 2(11.1.2.3) Note: A separate WebLogic 10.3.6 installation is required for Oracle Access Manager 11g Release 2(11.1.2.3) ▪ Oracle Access Manager Agent (WebGate) 11g Release 2 (11.1.2.3) ▪ Oracle Identity Management 11g Release 1 (11.1.1.9) Other components: <ul style="list-style-type: none"> ▪ Oracle BI Publisher 12c (12.2.1.2)

Check Supported Web Browser and Client Requirements

General requirements for PC clients running RWMS include the following.

Requirement	Version
Operating system	Windows 7, 10 Note: Oracle Retail assumes that the retailer has ensured its Operating System has been patched with all applicable Windows updates.
Terminal Server	Windows Server 2012R2
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 Mozilla Firefox 52+ ESR

Supported Mobile Client Device Requirements

The RWMS 15.0.2 Mobile user interface (Activity Monitoring, Labor Management, and Appointment Calendar) is supported on iPads running iOS 9. Please refer to the Oracle ADF 12.2.1.2 supported browser matrix available on Oracle Technology Network for more details on mobile browser support.

Note: The Oracle ADF 12.2.1.2 Browser Certification Matrix is available at:
<http://www.oracle.com/technetwork/developer-tools/jdev/documentation/122-cert-2708091.html>

Supported Radio Frequency Device Requirements

Minimum requirements for radio frequency devices in order to run the RWMS application are:

- Minimum RF Screen Sizes
- Hand held: 240w x 320h pixels
- Wrist mount: 320w x 240h pixels
- Truck mount - half screen: 800w x 320h pixels

Software Required on Handhelds

- Remote Desktop Client (aka Microsoft Terminal Services Client)
- DataWedge (software provided by and maintained by Motorola for use with the barcode scanners)

Note: The requirements above are based on the testing that was done using the following:

- RF devices running on Windows CE 5.0
 - Motorola VC5090 – (Truck Mount)
 - Motorola WT4090 – (Wrist mount)
 - Symbol MC9090 – (Hand Held)
 - Motorola WT4070 – (Wrist mount)
 - RF devices running on Windows CE 6.0:
 - Honeywell Thor VM1D – (Truck Mount)
 - Honeywell MX7T – (Hand Held)
-

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)	15.0.2
Oracle Retail Store Inventory Management (SIM)	15.0.2

Supported Oracle Retail Integration Technologies

Integration Technology	Version
Oracle Retail Integration Bus (RIB)	15.0.2
Oracle Retail Service Backbone (RSB)	15.0.2

UNIX User Account Privileges to Install the Software

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

For example, oretail.

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

RAC and Clustering

The Oracle Retail Warehouse Management System 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 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. Forms connections will continue to use OCI connections.

Clustering for WebLogic Server 10.3.6 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 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, 12c (12.2.1.0.0) Part Number E56928-01
- Oracle Real Application Clusters Administration and Deployment Guide 12c Release 1 (12.1) E48838-10

Database Installation Tasks – Full

This chapter describes the tasks required for a full database installation.

Note: If RWMS 15.0.1 software is already installed, please see [“Database Installation Tasks – Patch”](#) for information on Patching to RWMS 15.0.2.

If a database has been created for Oracle Retail Merchandising System (RMS) 15.0.2 and you would like the Retail Warehouse Management System (RWMS) to be in the same database, you can skip this step and advance to the [Create the Schema Owner for RWMS](#) section of this guide. However, if you want RWMS to reside in a dedicated database, follow the steps below to create a new one.

Create Staging Directory for RWMS Installer

To create the staging directory for RWMS installer, complete the following steps.

Note: The same installer can be used to install multiple RWMS components. If you are installing any of the RWMS components (Database, Application, Enhanced Dashboards and Navigation) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the database server as a user that can connect to the RWMS database.
2. Create a staging directory for the RWMS installation software.
3. Copy the `rwms15installer.zip` file from the RWMS 15.0.2 release to the staging directory. This is referred to as `STAGING_DIR` when installing database software.
4. Change directories to `STAGING_DIR` and extract the `rwms15installer.zip` file. This creates an `rwms/installer/` subdirectory under `STAGING_DIR`.

Create the RWMS Database

It is assumed that Oracle Enterprise Edition 12c Release 1, with all appropriate patches, has already been installed. If not, see [Check Supported Database Server Requirements](#) in Chapter 1 before proceeding. Additionally, `STAGING_DIR` in this section refers to the directory created in the steps for “Create Staging Directory for RWMS Installer” in Chapter 1.

If a database has already been created, it is necessary to review the contents of this section to determine if all database components have been installed and configured properly. See also [Appendix: Oracle Database 12cR1 Parameter File](#).

If a database instance has not been created, create one using database creation templates via DBCA in silent mode.

Create the Database Instance Using the Oracle Generic Template

Prerequisites

- The 12.1.0.2 binary must have already been installed. Refer to the Database Server Preinstallation section for all the required oneoff patches.

Background

As of 14.1, Oracle Retail no longer delivers custom database template files. Instead, databases can be created using the generic, Oracle-delivered templates located in the \$ORACLE_HOME/assistants/dbca/templates directory.

```
$ /<ORACLE_HOME>/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=<Oracle Home Location>
$export ORACLE_BASE=<ORACLE BASE LOCATION>
$export PATH=$ORACLE_HOME/bin:$PATH
$ cd into $ORACLE_HOME/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
ALL6UTF16 -redoLogFileSize 100 -initParams nls_date_format=DD-MON-
RR,nls_language=AMERICAN,nls_calendar=GREGORIAN,fast_start_mttr_target=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] to [-createAsContainerDatabase false].

3. Execute the following commands 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.

Create the Required RWMS Tablespaces

Release 15.0.2 uses table spaces RETAIL_DATA, RETAIL_INDEX, ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX.

Note: If you have created tablespaces during the implementation or upgrade of the Retail Merchandising System 15.0.2, you are no longer required to create another set of tablespaces for the Warehouse Management System as they use the same ones.

The ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX tablespaces hold data which may include Personally Identifiable Information data (PII Data). If you hold an Advanced Security Option license, you can choose to create these two tablespaces with TDE tablespace encryption to protect the PII data at rest. If you do not hold an Advanced Security Option license, you can create the tablespaces as normal tablespaces. The tablespace names must always be ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX regardless of whether TDE encryption is used, because the table and index creation scripts look for these specific names.

1. Modify STAGING_DIR/rwms/installer/create_db/create_wms_tablespaces.sql. The table below shows the default initial sizes.
2. Once this script has been modified, execute it in SQL*Plus as sys.
3. Review create_wms_tablespaces.log for errors and correct as needed.
4. If you do not wish to use TDE tablespace encryption:
 - a. Modify STAGING_DIR/rwms/installer/create_db/create_encrypted_tablespaces_no_TDE.sql.
 - b. Run the script using SQL*Plus as sys.
 - c. Review Create_encrypted_retail_tablespaces_no_TDE.log for errors and correct as needed.
5. If you hold an Advanced Security Option license and wish to use TDE tablespace encryption:
 - a. Modify STAGING_DIR/rwms/installer/create_db/create_encrypted_tablespaces_TDE.sql.
 - b. Run the script using SQL*Plus as sys.
 - c. Review Create_encrypted_retail_tablespaces_TDE.log for errors and correct as needed.
 - d. Refer to [Appendix: Tablespace Creation Scripts](#) for details about how to create tablespaces in an encrypted format.

The standard tablespace scripts contain the DDL for creating the required tablespaces which can extend up to the following sizes:

TABLESPACE_NAME	SIZE
ENCRYPTED_RETAIL_INDEX	12G
ENCRYPTED_RETAIL_DATA	10G
RETAIL_INDEX	10G

TABLESPACE_NAME	SIZE
RETAIL_DATA	8G
LOB_DATA	2G
USERS	2G

These sizes are sufficient for the initial installation of RWMS database objects. Although using the initial values is not recommended for a production environment, it is possible to use them for the purpose of creating a small test environment. For additional assistance with production database sizing, contact Oracle Retail Services.

Create the Schema Owner for RWMS

Create an Oracle schema that will own the RWMS application. Also, create RWMS ADF and RUNTIME users. Refer to the section below.

Note: The RWMS schema owner must be created prior to running the RWMS database schema installation. The installer will validate this user before proceeding with installation.

1. Change directories to STAGING_DIR/rwms/installer/create_db.
2. The create_user script relies on an empty role, developer, being created. Log into sqlplus as sysdba, connect to the respective PDB and run the following command to create that role:
SQL> create role developer;
3. Enter the following command to create the schema owner:
SQL> @create_user.sql
The following prompts will appear:
 - Schema Owner – the Oracle user that will own all RWMS objects. Referred to in this document as RWMS15DEV.
 - Password – the password associate with RWMS15DEV.
 - Temp Tablespace – the temporary tablespace for RWMS15DEV.
4. Check the log file create_<Schema_Owner>.lst for any errors. This log file should be removed to prevent the password from being compromised.
5. Create rwms_adf_user. Use the below script to create rwms_adf_user
SQL> @create_rwms_adf_user.sql
 - Password – enter the password for the user rwms_adf_user
 - Temp Tablespace – the temporary tablespace for the user
6. Create runtime user and role. Use the below script to create runtime user and the role.
SQL >@create_runtime_user_role.sql
 - runtime_user – enter the database user name for running RWMS application
 - runtime_password – enter the password for the user
 - Temp Tablespace – the temporary tablespace for the user

Run the RWMS Database Schema Installation

To run the RWMS database schema installation, complete the following steps.

Note: See [Appendix: RWMS Database Schema Installation Screens](#) for details on the RWMS Database Schema installation screens and fields in the installer.

1. Change directories to STAGING_DIR/rwms/installer.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc)

Example: prompt\$. oraenv
ORACLE_SID = [] ? mydb
prompt\$

Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

Example: prompt\$ echo \$ORACLE_HOME
/u00/oracle/product/mydbversion
prompt\$ echo \$ORACLE_SID
mydb

3. Set and export the following environment variables. These variables are needed in addition to the environment variables set by the oraenv script above.

Variable	Description	Example
JAVA_HOME	Java home needed to run the GUI. Java 1.7 is required	JAVA_HOME=/usr/java/jdk1.7.0_17.64bit export JAVA_HOME
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
DISPLAY	Address and port of X server on desktop system of user running install. Optional for dbschema installer	DISPLAY=<IP address>:0 export DISPLAY

4. If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
5. If the installer has already been run in this location you may wish to back up the ant.install.properties file.
6. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

./install.sh [text | silent]

7. Verify that the installer reports "SUCCESS" for the Database Preinstall Check. If it reports "FAILED," verify that your environment variables are set properly.
8. Check the Install Database Objects checkbox and continue with installer.
9. After the installer is complete, you can check its log file:
STAGING_DIR/rwms/installer/logs/rwms-install.<timestamp>.log.

Note: The installer leaves behind the `ant.install.properties` file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

```
chmod 600 ant.install.properties
```

10. After the database installation is completed, run the `base_dms_user.sql` script to create the super user required to log in to the RWMS application.
 - Change directories to `STAGING_DIR/rwms/installer/rwms15/Install_Data/Source`
 - Log in to the database as the RWMS schema owner
 - `SQL> @base_dms_user.sql`

Example:

```
*****
```

If User already exists in the Facility, New Password entered below will be updated

```
*****
```

Please enter the Superuser name for RWMS: par3214

Please enter the Superuser User Id for RWMS: par3214

Please enter the Superuser password: <retail>

Database Installation Tasks – Patch

The RWMS 15.0.2 installer may be used to apply the RWMS patch. Before you apply the RWMS 15.0.2 patch:

- Make a backup of all your objects and database schema.
- Review the RWMS 15.0.2 Release Notes.
- Review each of the enclosed defect documents.

Create Staging Directory for RWMS Installer

To create the staging directory for RWMS installer, complete the following steps.

Note: The same installer can be used to patch multiple RWMS components. If you are installing any of the RWMS components (Database, Application, and Enhanced Navigation and Dashboards) on the same server as the same user, they can use the same installer and this step does not need to be repeated.

1. Log into the database server as a user that can connect to the RWMS database.
2. Create a staging directory for the RWMS installation software.
3. Copy the `rwms15installer.zip` file from the RWMS 15.0.2 release to the staging directory. This is referred to as `STAGING_DIR` when installing database software.
4. Change directories to `STAGING_DIR` and extract the `rwms15installer.zip` file. This creates an `rwms/installer/` subdirectory under `STAGING_DIR`.

(Optional) Analyze Changes in the Patch

Note: See [Appendix: RWMS Analyze Tool](#) for details and instructions to run the RWMS Analyze Tool. This appendix also contains screens and fields in the tool.

Run the RWMS Database Schema Installation

To run the RWMS database schema installation, complete the following steps.

Note: See [Appendix: RWMS Database Schema Installation Screens](#) for details on the RWMS Database Schema installation screens and fields in the installer.

1. Change directories to `STAGING_DIR/rwms/installer`.
2. Source the `oraenv` script to set up the Oracle environment variables (`ORACLE_HOME`, `ORACLE_SID`, `PATH`, etc).

Example:

```
prompt$ . oraenv
ORACLE_SID = [] ? mydb
prompt$
```

Verify the `ORACLE_HOME` and `ORACLE_SID` variables after running this script.

```

Example: prompt$ echo $ORACLE_HOME
              /u00/oracle/product/mydbversion
prompt$ echo $ORACLE_SID
              mydb

```

- Set and export the following environment variables. These variables are needed in addition to the environment variables set by the oraenv script above.

Variable	Description	Example
JAVA_HOME	Java home needed to run the GUI. Java 1.7 is required	JAVA_HOME=/usr/java/jdk1.7.0_17.64bit export JAVA_HOME
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
DISPLAY	Address and port of X server on desktop system of user running install. Optional for dbschema installer	DISPLAY=<IP address>:0 export DISPLAY

- If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
- If the installer has already been run in this location you may wish to back up the ant.install.properties file.
- Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh [text | silent]
```

- Verify that the installer reports "SUCCESS" for the Database Preinstall Check. If it reports "FAILED," verify that your environment variables are set properly.
- Check the Install Database Objects checkbox and continue with installer.
- On the RETAIL_HOME screen, select the RETAIL_HOME of your previous installation.
- On the Wallet password screen, enter the wallet password you used in the previous installation.
- After the installer is complete, you can check its log file:
STAGING_DIR/rwms/installer/logs/rwms-install.<timestamp>.log.

Note: The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

```
chmod 600 ant.install.properties
```

Application Server Installation Tasks – RWMS Forms

It is assumed that WebLogic 12c (12.2.1.2) with Forms 12c (12.2.1.2) has already been installed. If not, refer to “[Check Application Server Requirements](#)” in Chapter 1, “[Preinstallation Tasks](#)” before proceeding.

Note: Oracle installation instructions for Oracle Forms 12c can be found at <https://docs.oracle.com/middleware/12212/formsandreports/install-fnr/toc.htm>

The application installer will install and compile the various Forms dependencies and validate them against the RWMS Database Schema installation performed in the previous chapter. Installation of BI Publisher server components are covered in Chapter 8

Note: The guide does not make recommendations about the clustering or high availability configurations for WebLogic as these will largely be dependent on your fault-tolerance and scalability requirements.

Note: \$ORACLE_HOME/network/admin/tnsnames.ora file must be configured in this WebLogic installation. Forms and reports use this information for connectivity.

Prepare Application Server for RWMS

To prepare the application server for RWMS, complete the following steps.

- Create and configure tnsnames.ora file
- Configure Tk2Motif.rgb
- Create Staging Directory for RWMS Installer

Note: ORACLE_HOME is the location where Oracle Forms 12c has been installed.
For Example, ORACLE_HOME=
/u00/webadmin/products/wls_forms
ORACLE_INSTANCE is the instance that is created during installation of Oracle Forms 12c and contains the executables to compile forms. For Example, ORACLE_INSTANCE location like
/u00/webadmin/config/domains/wls_forms/ClassicDomain/config/fmwconfig/components/FORMS/instances/formms1

Create and Configure tnsnames.ora file

A tnsnames.ora file must be configured and placed in the \$ORACLE_HOME/network/admin/directory in order for Oracle Forms to have connectivity to the database. This database connectivity is used for compiling Oracle Forms at installation time. See Appendix: Tablespace Creation Script for an example setup of the tnsnames.ora file.

Configure Tk2Motif.rgb

To create the staging directory for the RWMS Installer, complete the

1. The Tk2Motif.rgb file that is sent out with WebLogic (12.2.1.2) must be modified. It is located in the following directory:
<ORACLE_HOME>/frcommon/templates/tk/config/US/
2. Make a copy of the file Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).
3. Modify the file Tk2Motif.rgb file so that it contains the following line:
Tk2Motif*fontMapCs: iso8859-2=AL32UTF8
4. Copy the file to the following directory:
<ORACLE_HOME>/guicommon/tk/admin/Tk2Motif.rgb.

Create Staging Directory for RWMS Installer

To create the staging directory for the RWMS Installer, complete the following steps.

Note: The same installer can be used to install multiple RWMS components. If you are installing any of the RWMS components (Database, Application, Enhanced Dashboards and Navigation) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the application server as a user with read and write access to the WebLogic files.
2. Create a staging directory for the RWMS installation software.
3. Copy the file rwms15installer.zip from the RWMS 15.0.2 release to the staging directory. This will be referred to as STAGING_DIR when installing application software and reports.
4. Change directories to STAGING_DIR and extract the file rwms15installer.zip. This will create an rwms/installer subdirectory under STAGING_DIR.

Install RWMS Oracle Forms

To install RWMS Oracle forms, complete the following steps:

Note: See [Appendix: RWMS Application \(forms\) Installation Screens](#) for details about the RWMS Forms application screens and fields in the installer.

1. Log on to your application server as the user that has write access to WebLogic.
2. Change directories to STAGING_DIR/rwms/installer. Set and export the following environment variables.

Variable	Description	Example
DOMAIN_HOME	The location of the WebLogic domain that contains Oracle Forms.	DOMAIN_HOME= /u00/webadmin/config/domains/wls_forms/ClassicDomain export DOMAIN_HOME
WLS_INSTANCE	The name of the managed server that the forms component is installed in.	WLS_INSTANCE=WLS_FORMS export WLS_INSTANCE
ORACLE_SID	The database/SID or pluggable database service name where the RWMS schema resides.	ORACLE_SID=rwmsdb
ORACLE_HOME	The location of the WebLogic installation that contains the Oracle Forms domain.	ORACLE_HOME= /u00/webadmin/products/wls_forms/ export ORACLE_HOME
ORACLE_INSTANCE	This is the directory path in which the Oracle Forms instance resides.	ORACLE_INSTANCE= /u00/webadmin/config/domains/wls_forms/ClassicDomain/config/fmwconfig/components/FORMS/instances/forms1 export ORACLE_INSTANCE
DISPLAY	Address and display number of X server on desktop system of user running install. Required for forms application installer, even when the installer is run in text mode.	DISPLAY=<IP address>:0 export DISPLAY
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
JAVA_HOME	JAVA_HOME should point to your valid JDK installation. Typically, this is the JDK being used by WEBLOGIC HOME.	JAVA_HOME=/vol.rtk/java/os_platform/jdk1.8.64bit

- Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

```
./install.sh
```

Depending on system resources, a typical installation takes anywhere from 45 minutes to two hours.

The installer asks for an installation directory. This is the destination directory for the RWMS files. This directory will be referred to as RETAIL_HOME for the remainder of this chapter. Do not provide a RETAIL_HOME that is located at or underneath STAGING_DIR.

- After the installation is complete, you can check its log file: STAGING_DIR/rwms/installer/logs/rwms-install.<timestamp>.log. The directory

RETAIL_HOME/orpatch/logs/detail_logs/oraforms_rwms/ will contain log files for specific forms, with information about possible failed compilations.

5. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example:

```
chmod 600 ant.install.properties
```

6. If during the screens you chose not to have the installer automatically configure WebLogic, after the installation is complete the files that would have been updated by the installer will be located in a directory and will have to manually be moved to correct directories as specified by the installer output like the example below.

Example:

```
#####
                          Weblogic Configuration Tasks
#####
Contact your Weblogic administrator and have them make backups of the
following files:

/u00/webadmin/config/domains/wls_forms/ClassicDomain/config/fmwconfig/servers/
WLS_FORMS/applications/formsapp_12.2.1/config/oracle/forms/registry/Registry.d
at
/u00/webadmin/products/wls_forms/forms/java/extensions.jnlp
/u00/webadmin/config/domains/wls_forms/ClassicDomain/config/fmwconfig/servers/
WLS_FORMS/applications/formsapp_12.2.1/config/formsweb.cfg

Have the Weblogic administrator stop WLS_FORMS,
copy everything in /u00/oretail/rwms15.0.2/tst/post/MW_HOME
to /scratch/u00/webadmin/products/wls_forms/wlserver/..
and copy everything in /u00/oretail/rwms15.0.2/tst/post/DOMAIN_HOME
to /u00/webadmin/config/domains/wls_forms/ClassicDomain
to update the files
and then start WLS_FORMS
for the changes to take effect.

example:
cd /u00/oretail/ rwms15.0.2/tst/post/MW_HOME
cp -R * /scratch/u00/webadmin/products/wls_forms/wlserver/..
cd /u00/oretail/rwms15.0.2/tst/post/DOMAIN_HOME
```

```
cp -R * /u00/webadmin/config/domains/wls_forms/ClassicDomain
```

Application Server Installation Tasks – Patch

The RWMS 15.0.1 installer may be used to apply the RWMS patch. Before you apply the RWMS 15.0.1 application patch:

- Review the RWMS 15.0.1 Release Notes.

Create Staging Directory for RWMS Installer

To create the staging directory for the RWMS Installer, complete the following steps.

Note: The same installer can be used to install multiple RWMS components. If you are installing any of the RWMS components (Database, Application, Enhanced Dashboards and Navigation) on the same server, they can use the same installer and this step does not need to be repeated.

1. Log into the application server as a user with read and write access to the WebLogic files.
2. Create a staging directory for the RWMS installation software.
3. Copy the file `rwms15installer.zip` from the RWMS 15.0.1 release to the staging directory. This will be referred to as `STAGING_DIR` when installing application software and reports.
4. Change directories to `STAGING_DIR` and extract the file `rwms15installer.zip`. This will create an `rwms/installer` subdirectory under `STAGING_DIR`.

(Optional) Analyze Changes in the Patch

Note: See [Appendix: RWMS Analyze Tool](#) for details and instructions to run the RWMS Analyze Tool. This appendix also contains screens and fields in the tool.

Run the RWMS Application Installation

Note: See [Appendix: RWMS Application Installer Screens](#) for details about the RWMS application screens and fields in the installer.

Install RWMS Oracle Forms

To install RWMS Oracle forms, complete the following steps:

Note: See [Appendix: RWMS Application \(forms\) Installation Screens](#) for details about the RWMS Forms application screens and fields in the installer.

1. Log on to your application server as the user that has write access to WebLogic.
2. Change directories to `STAGING_DIR/rwms/installer`.
3. Set and export the following environment variables.

Variable	Description	Example
DOMAIN_HOME	The location of the WebLogic domain that contains Oracle Forms.	DOMAIN_HOME= /u00/webadmin/config/domains/wls_forms/ClassicDomain export DOMAIN_HOME
WLS_INSTANCE	The name of the managed server that the forms component is installed in.	WLS_INSTANCE=WLS_FORMS export WLS_INSTANCE
ORACLE_SID	The database/SID or pluggable database service name where the RWMS schema resides.	ORACLE_SID=rwmsdb
ORACLE_HOME	The location of the WebLogic installation that contains the Oracle Forms domain.	ORACLE_HOME= /u00/webadmin/products/wls_forms/ export ORACLE_HOME
ORACLE_INSTANCE	This is the directory path in which the Oracle Forms instance resides.	ORACLE_INSTANCE= /u00/webadmin/config/domains/wls_forms/ClassicDomain/config/fmwconfig/components/FORMS/instances/forms1 export ORACLE_INSTANCE
DISPLAY	Address and display number of X server on desktop system of user running install. Required for forms application installer, even when the installer is run in text mode.	DISPLAY=<IP address>:0 export DISPLAY
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG
JAVA_HOME	JAVA_HOME should point to your valid JDK installation. Typically, this is the JDK being used by WEBLOGIC HOME.	JAVA_HOME=/vol.rtk/java/os_platform/jdk1.8.64bit

4. Run the install.sh script to start the installer.

Note: Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.

`./install.sh [text | silent]`

5. Verify that the installer reports "SUCCESS" for the Application preinstall check. If it reports "FAILED," verify that your environment variables are set properly.
6. Check the Install Application checkbox and proceed with the installation.
7. On the RETAIL_HOME screen, select the RETAIL_HOME of your previous installation.
8. On the Wallet password screen, enter the wallet password you used in the previous installation.

9. After the installation is complete, you can check its log file: STAGING_DIR/rwms/installer/logs/rwms-install.<timestamp>.log. The directory RETAIL_HOME/orpatch/logs/detail_logs/oraforms_rwms/ will contain log files for specific forms, with information about possible failed compilations.
10. The installer leaves behind the ant.install.properties file for future reference and repeat installations. This file contains inputs you provided. As a security precaution, make sure that the file has restrictive permissions.

Example:

```
chmod 600 ant.install.properties
```

Enhanced Navigation and Dashboards Installation

The Enhanced Navigation and Dashboard of RWMS provides a more modern web application GUI to the product that runs the RWMS Oracle forms. Along with a rich web design, the Enhanced Navigation and Dashboard brings with it new navigation features, search mechanisms, easy-to-use favorites.

Before proceeding, you must install Oracle WebLogic Server (12.2.1.2). The Enhanced Navigation and Dashboards application is deployed to a WebLogic Managed server within the WebLogic installation. It is assumed Oracle Database has already been configured and loaded with the appropriate Oracle Retail Warehouse Management System schemas for your installation. ADF 12.2.1.2 should also be installed on the WebLogic installation. Additionally, STAGING_DIR in this section refers to the directory created in “Create Staging Directory for RWMS Application Files” in Chapter 1.

Installing a separate domain is recommended. It can be called “RWMSDomain” (or something similar) and will be used to install the managed server. The ADF libraries should be extended to this domain and the Enterprise Manager should be deployed.

You can create a new WebLogic server and install ADF libraries on it or create a WebLogic domain on the WebLogic server that has the ClassicDomain deployed

Steps to Create the New Domain with ADF Libraries

Fusion Middleware Infrastructure and WebLogic Server 12c (12.2.1.2) Installation

Create a directory to install the WebLogic (this will be the ORACLE_HOME):

Example: `mkdir -p /u00/webadmin/products/wls_retail`

1. Set the ORACLE_HOME, JAVA_HOME and DOMAIN_HOME environment variables:
 - ORACLE_HOME should point to your WebLogic installation.
 - JAVA_HOME should point to the Java JDK 1.8+. This is typically the same JDK which is being used by the WebLogic domain where application is getting installed.

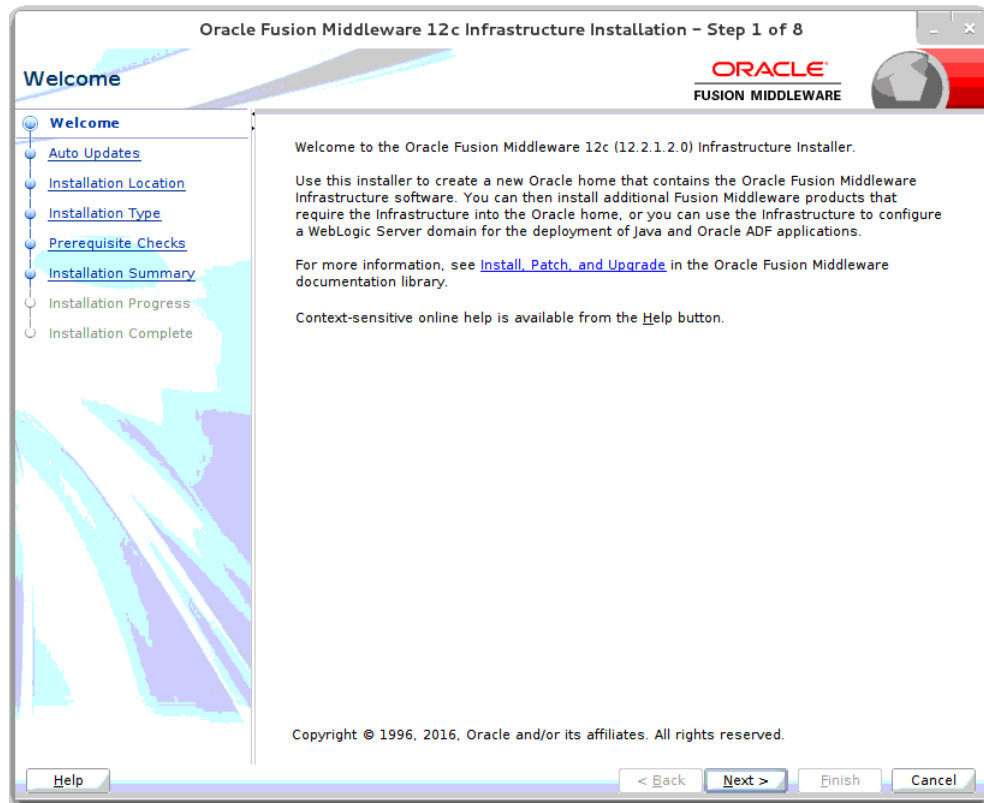
Example:

```
$export ORACLE_HOME=/u00/webadmin/products/wls_retail
$export JAVA_HOME=/u00/webadmin/products/jdk_java
(This should point to the Java which is installed on your server)
$export PATH=$JAVA_HOME/bin:$PATH
```

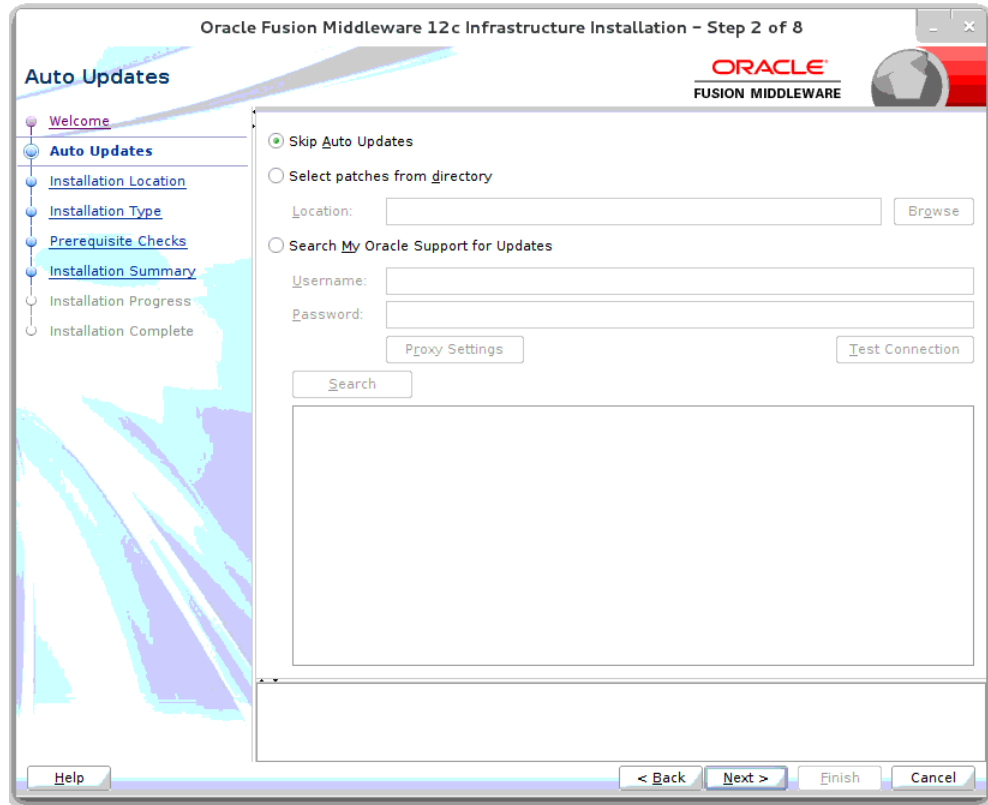
Going forward we will use the above references for further installations.

2. Go to location where the weblogic jar is downloaded and run the installer using the following command:
`java -jar ./fmw_12.2.1.2.0_infrastructure.jar`

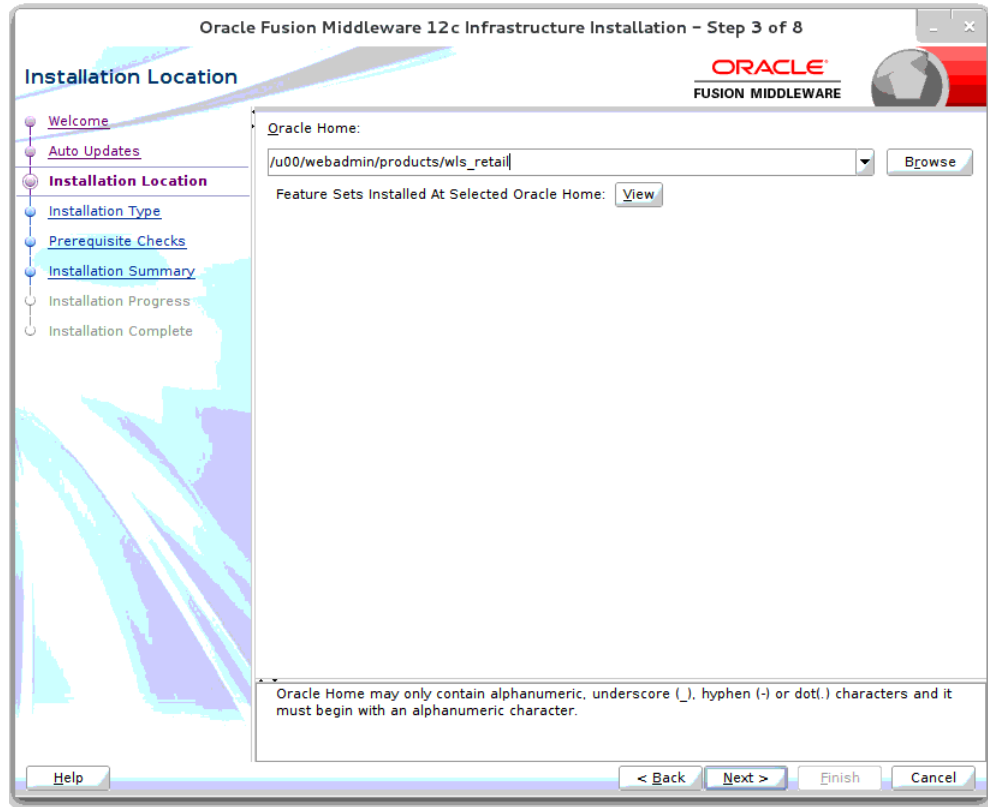
3. Welcome screen appears. Click **Next**.



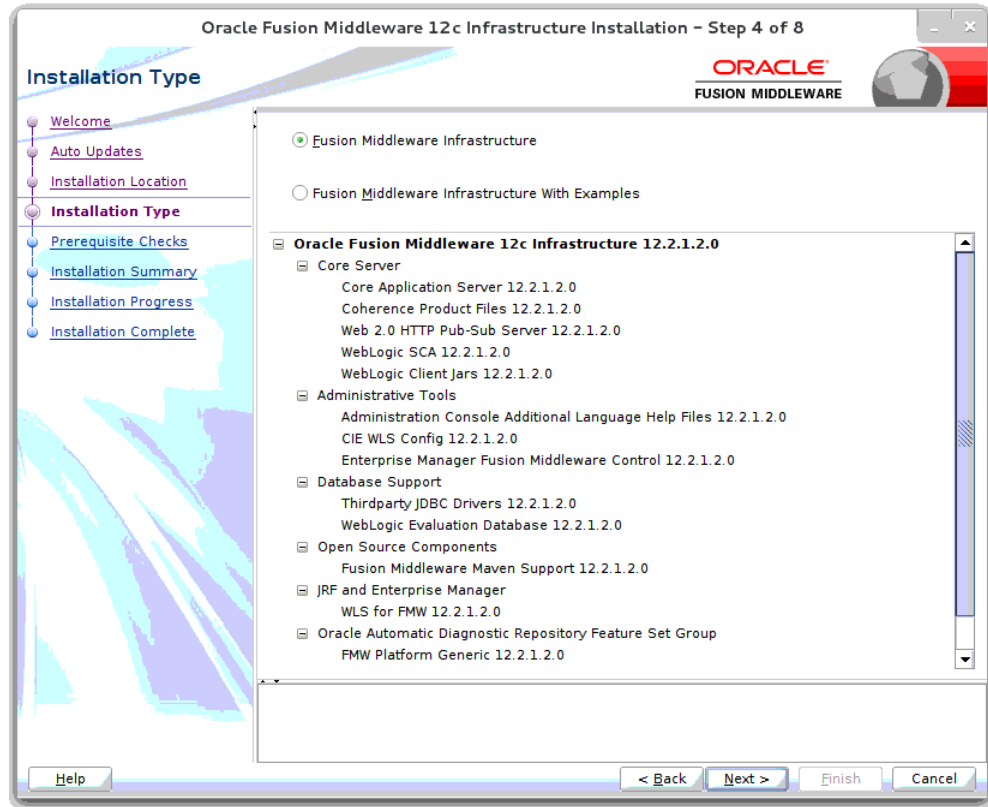
4. Click **Next**.



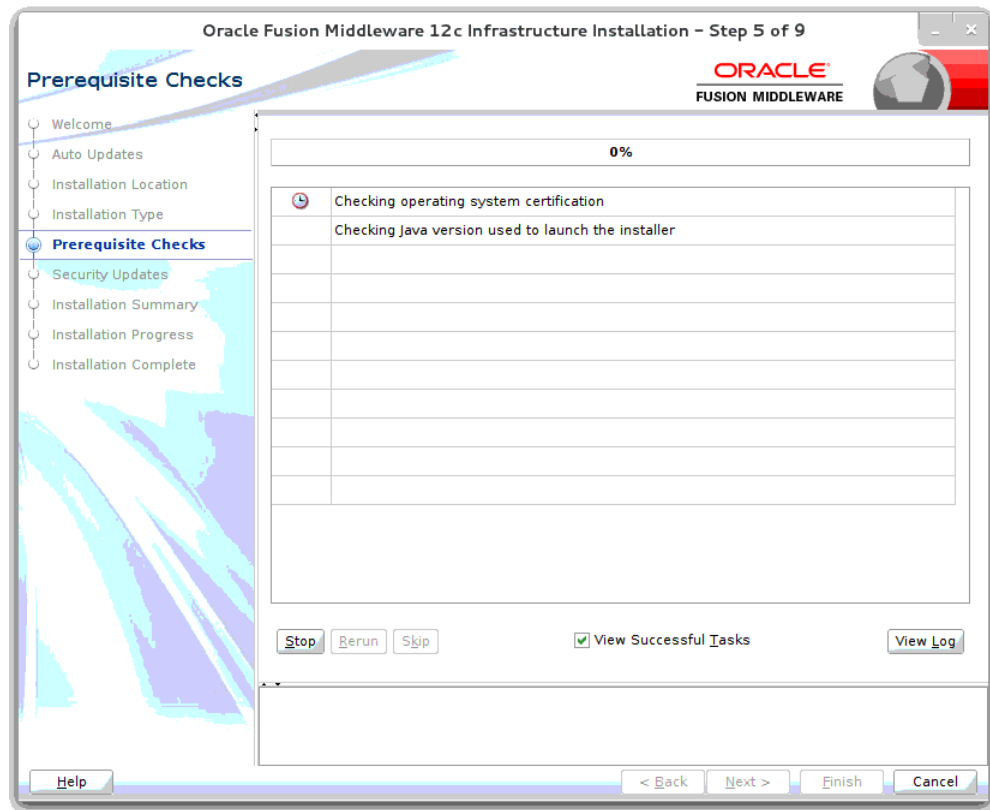
5. Enter the following and click **Next**.
Oracle home =<Path to the ORACLE_HOME>
Example:
/u00/webadmin/products/wls_retail



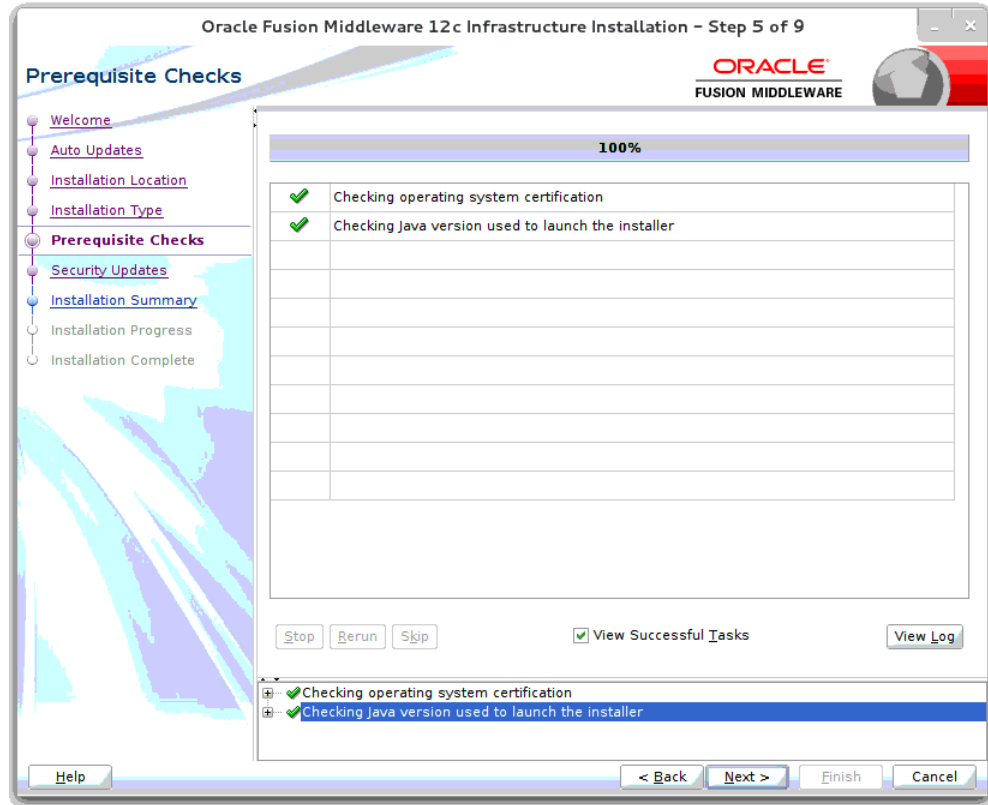
6. Select install type 'Fusion Middleware Infrastructure'. Click **Next**.



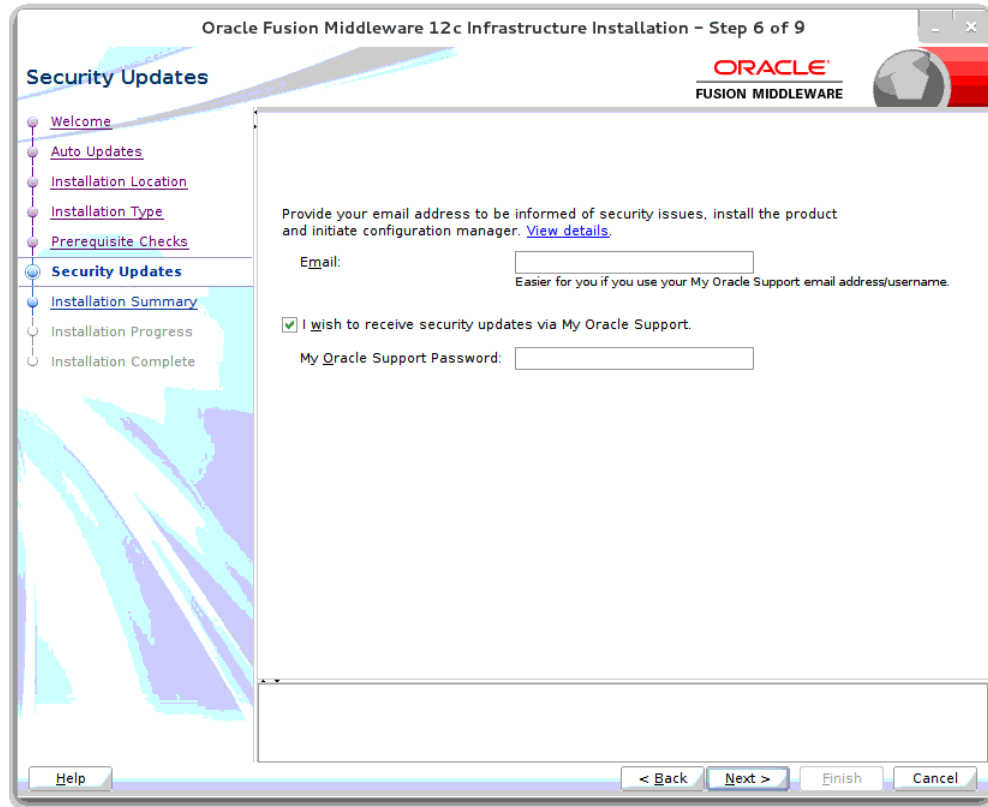
This screen will verify that the system meets the minimum necessary requirements.



7. Click Next.



8. If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates.
9. If you do not have one or if you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
10. Click **Next**.



11. Click Next.

Oracle Fusion Middleware 12c Infrastructure Installation - Step 6 of 9

Security Updates

ORACLE
FUSION MIDDLEWARE

Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Security Updates
Installation Summary
Installation Progress
Installation Complete

Provide your email address to be informed of security issues, install the product and initiate configuration manager. [View details.](#)

Email:
Easier for you if you use your My Oracle Support email address/username.

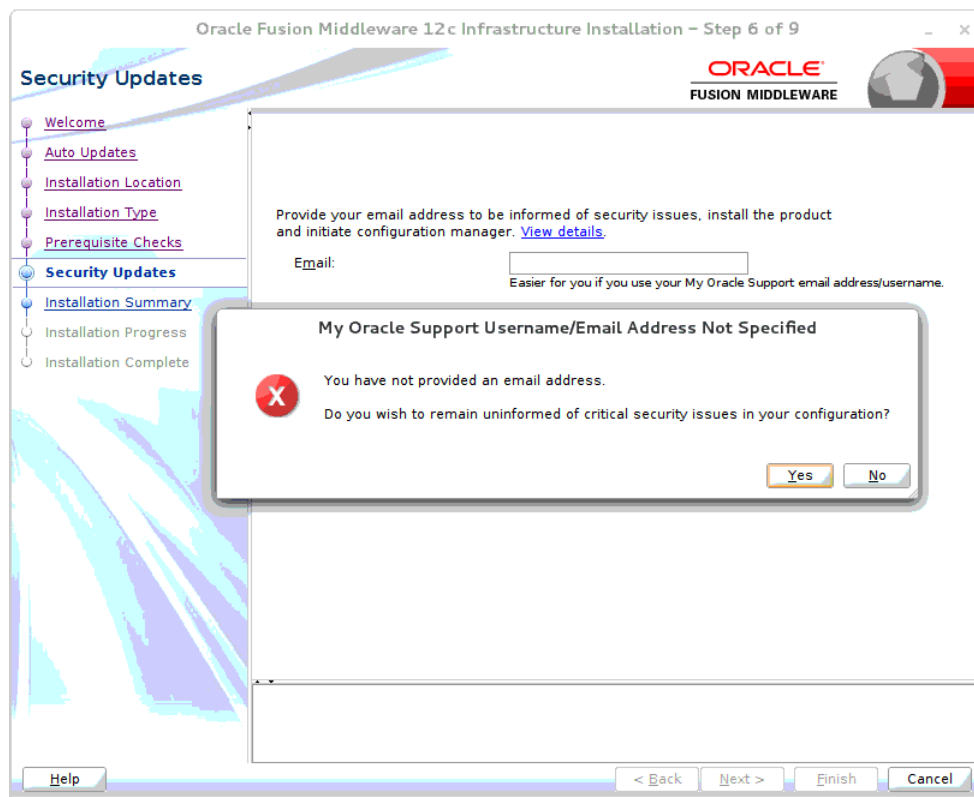
I wish to receive security updates via My Oracle Support.

My Oracle Support Password:

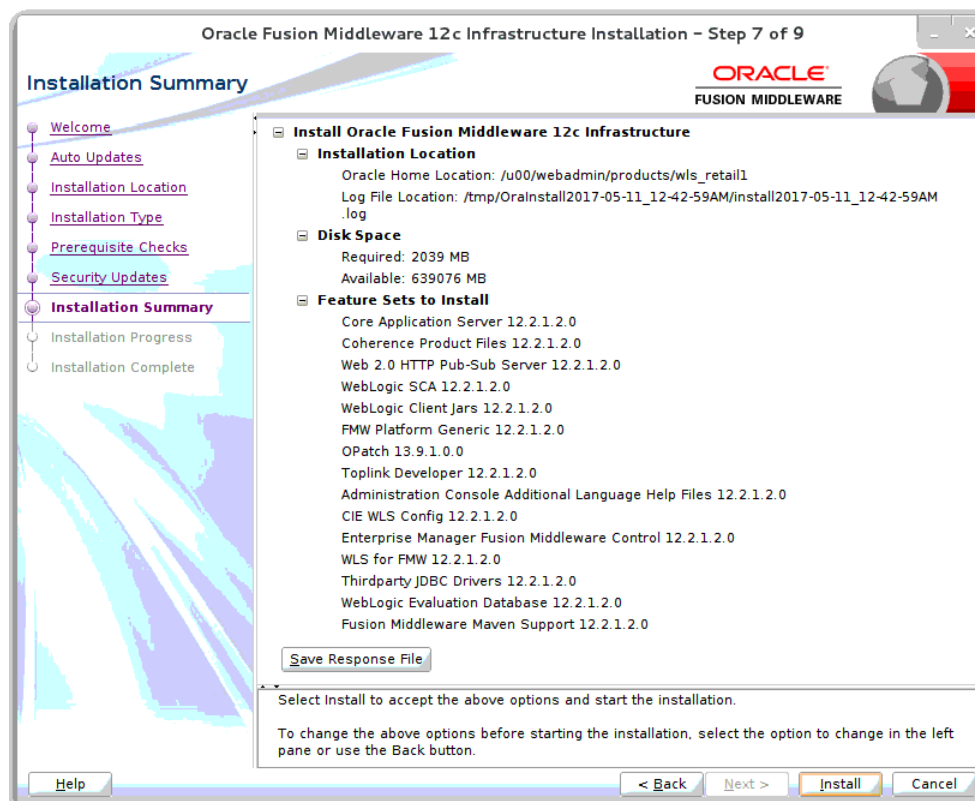
Help < Back Next > Finish Cancel

12. Click **Next**.

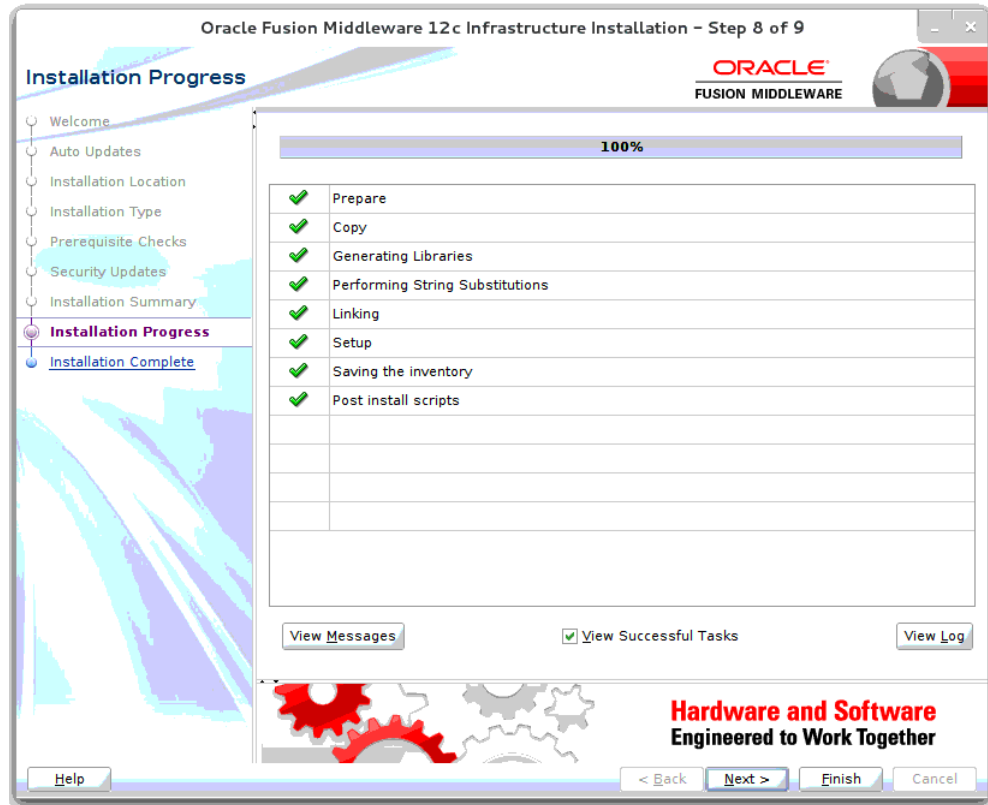
13. Click **Yes**, if you wish to remain uninformed of security issues in your configuration.



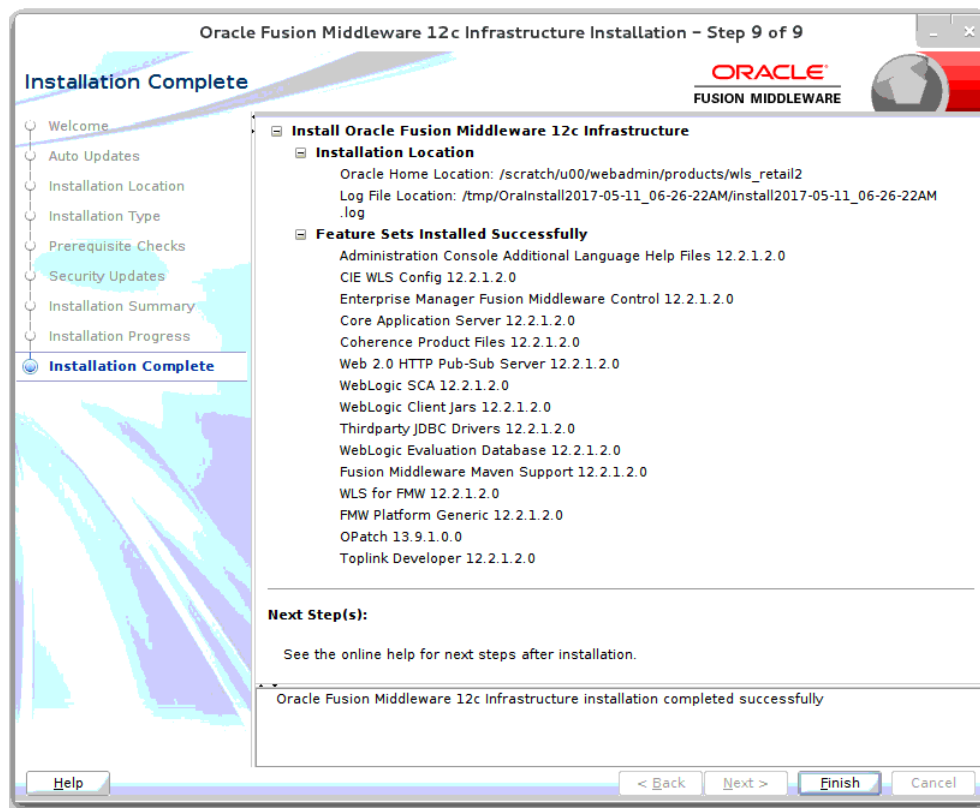
14. Click Install.



15. Click Next.



16. Click Finish.



Install RCU Database Schemas

The RCU database schemas are required for the installation of configuration of domain and retail application.

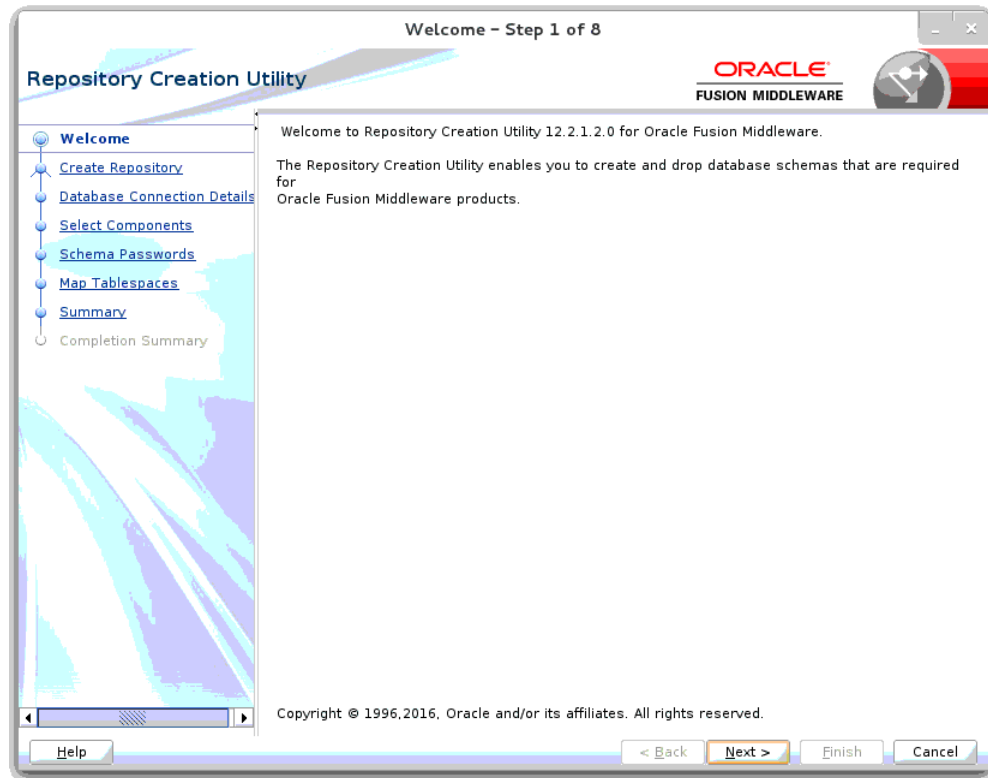
Note: Need user which have sys admin privileges to install the RCU database schemas.

The following steps are provided for the creation of the database schemas:

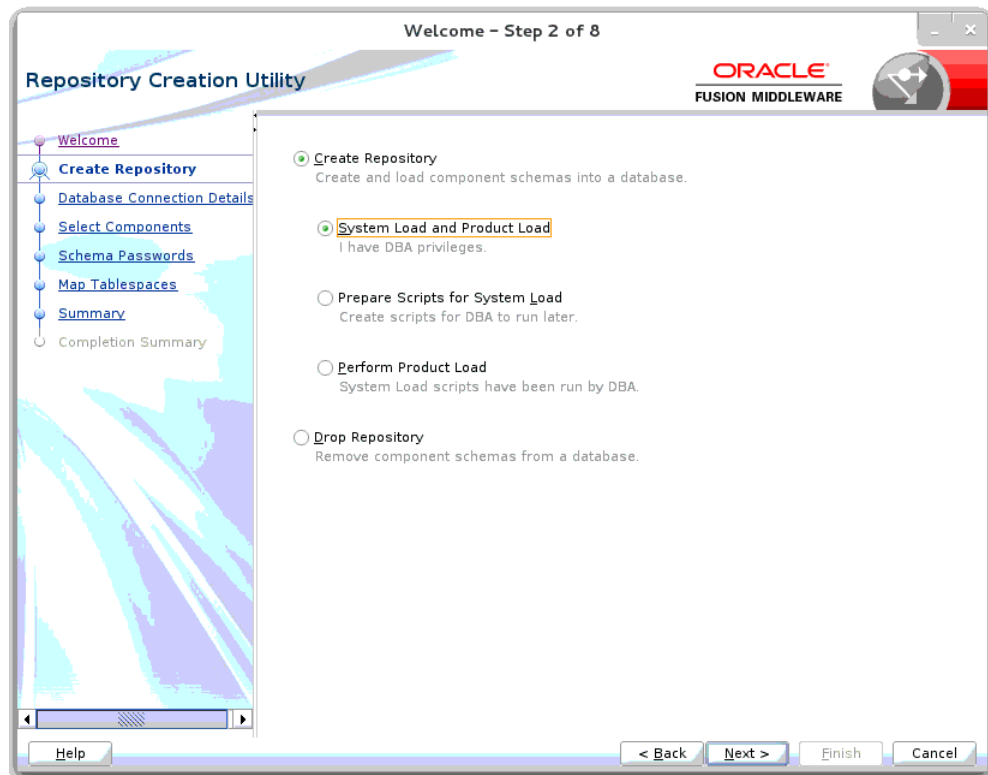
1. Navigate to the directory into which RCU is installed. For example:

```
<ORACLE_HOME>/oracle_common/bin/
```

 Run `"./rcu"`
2. Click **Next**.

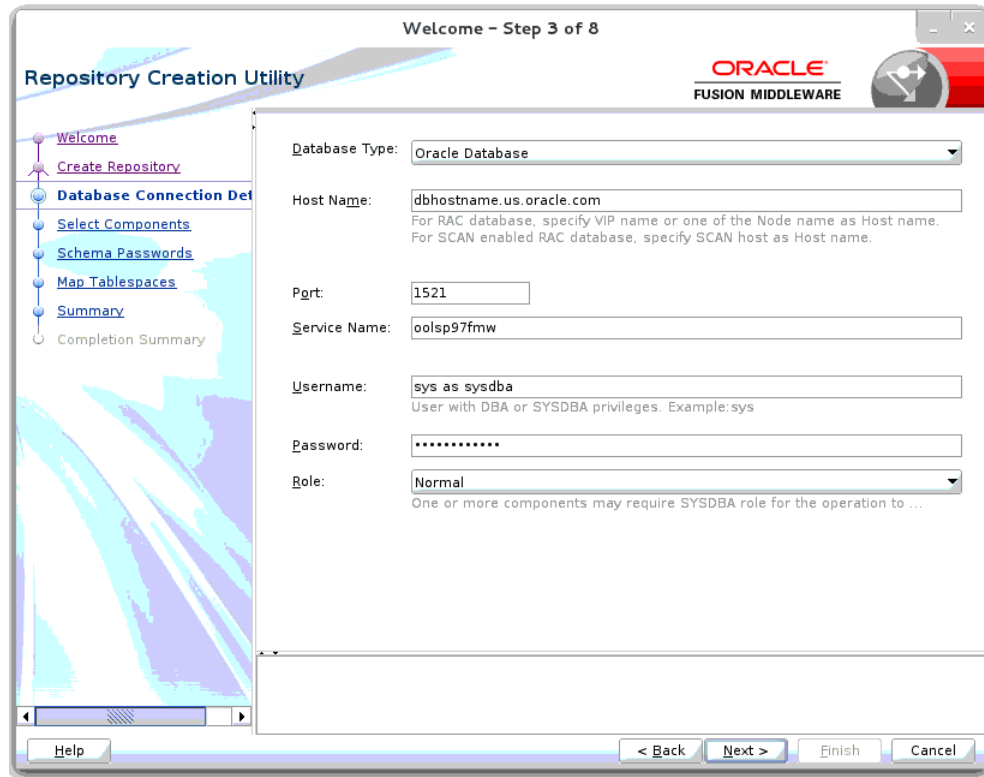


3. Select Create Repository and System Load and Product Load. Click Next.

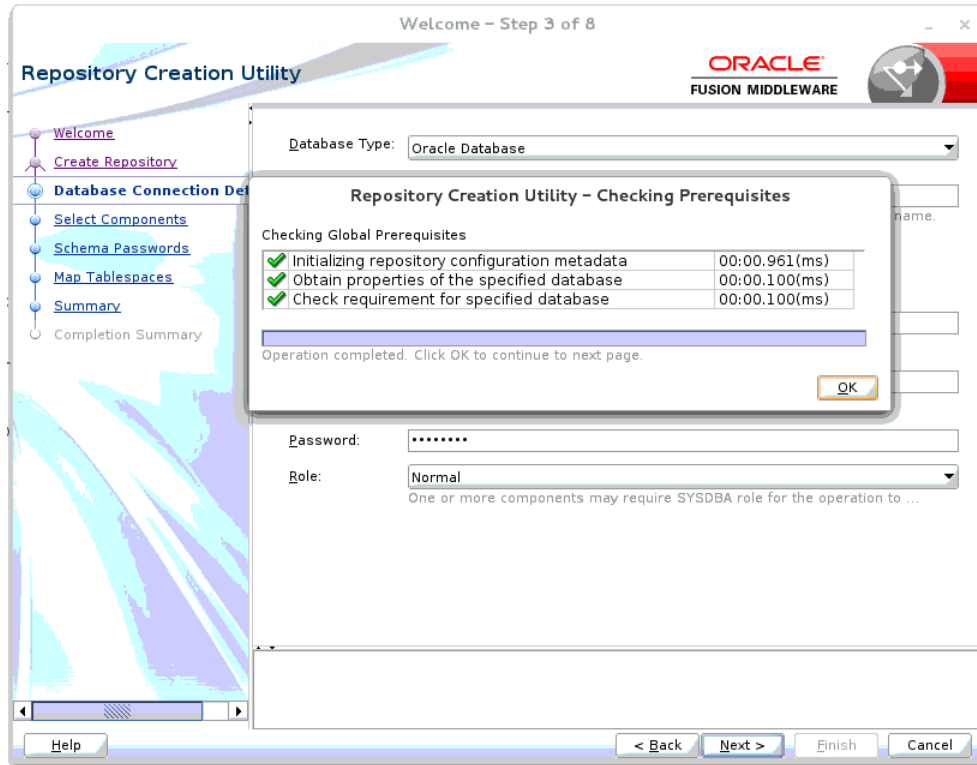


4. Enter database connection details:

- Database Type: Oracle Database
- Host Name: dbhostname.us.oracle.com
- Port: 1521
- Service Name: dbservicename
- Username: sys
- Password: <syspassword>
- Role: SYSDBA



5. Click **Next**. The Installer checks prerequisites.
6. When the prerequisite checks are complete, click **OK**. Click **Next**.



7. Click the **Create a new prefix** option, the prefix name for your schemas should be unique to your application environment.
Example: ReIM, ALLOC, ReSA, etc
8. Select the components to create:
 - Meta Data Services
 - Oracle Platform Security Services

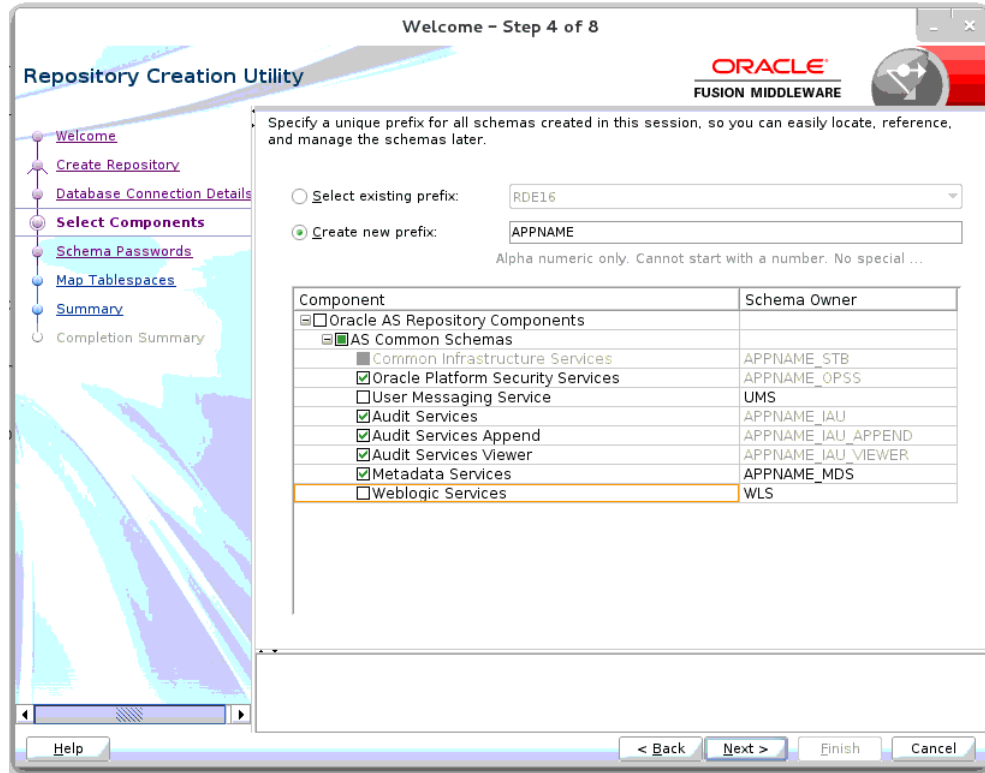
Note: Once OPSS schema is selected, the following dependent schemas will get selected automatically.

Audit Services

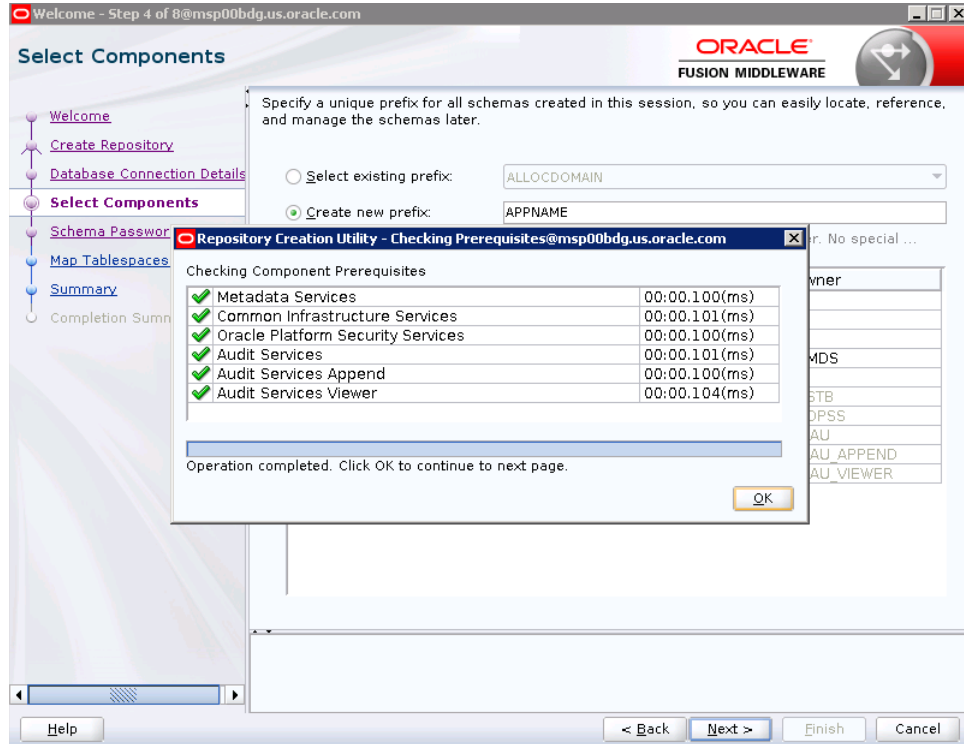
Audit Services Append

Audit Services Viewer

Note: STB schema will be already selected as part of the Common Infrastructure component.

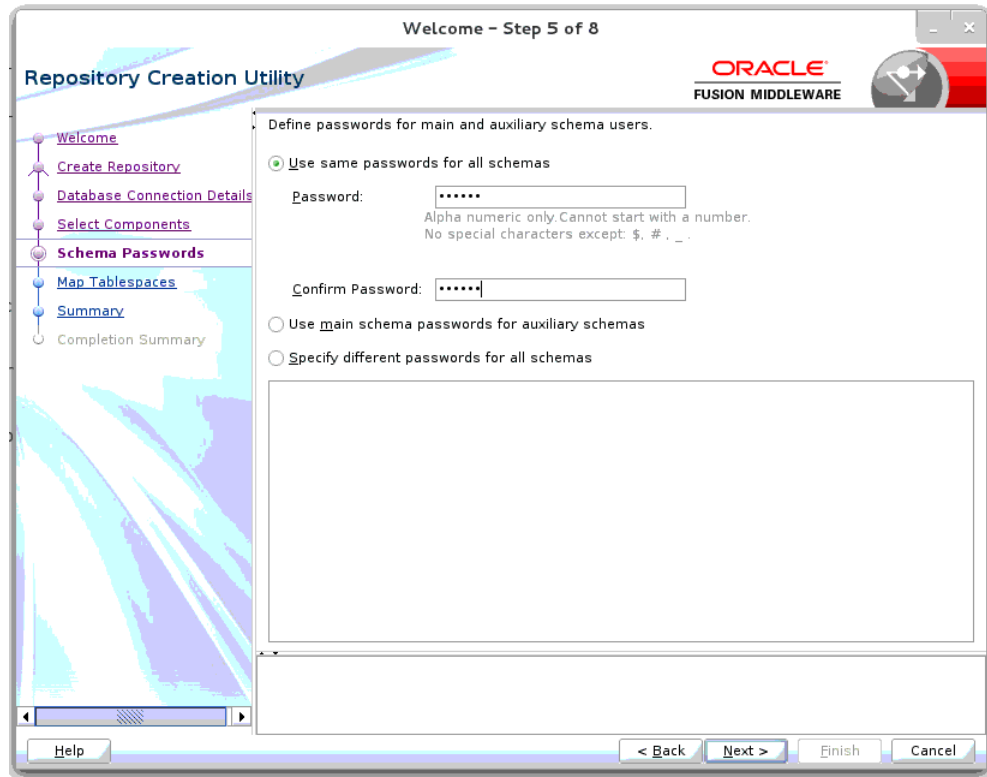


9. Click Next.

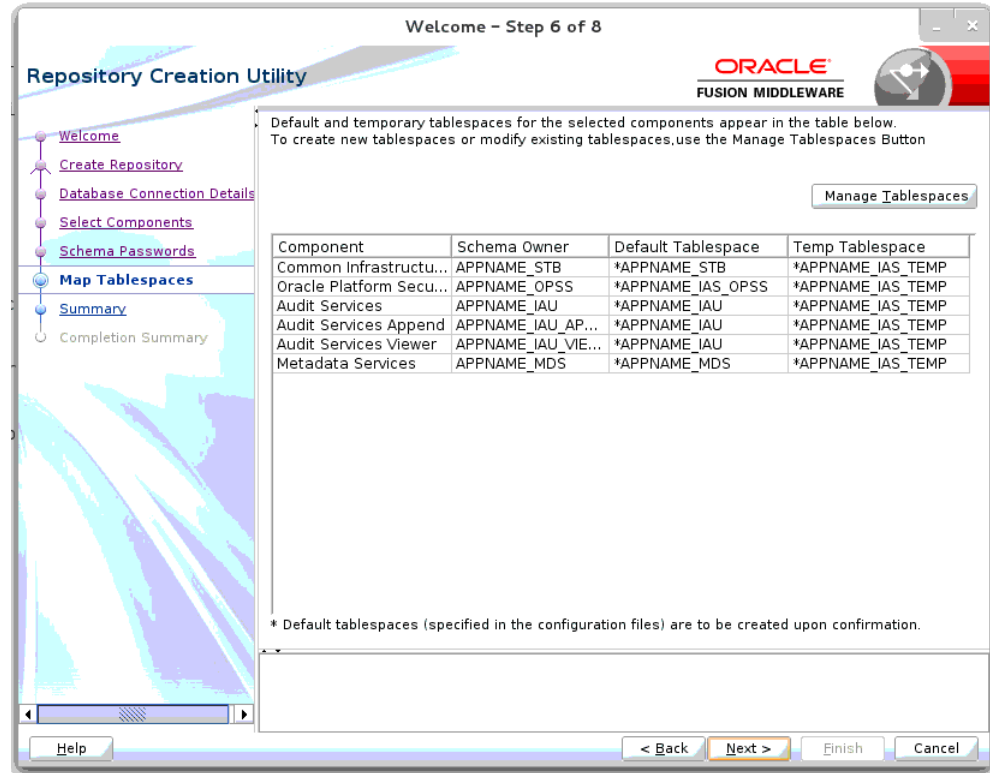


10. Enter a password of your choice.

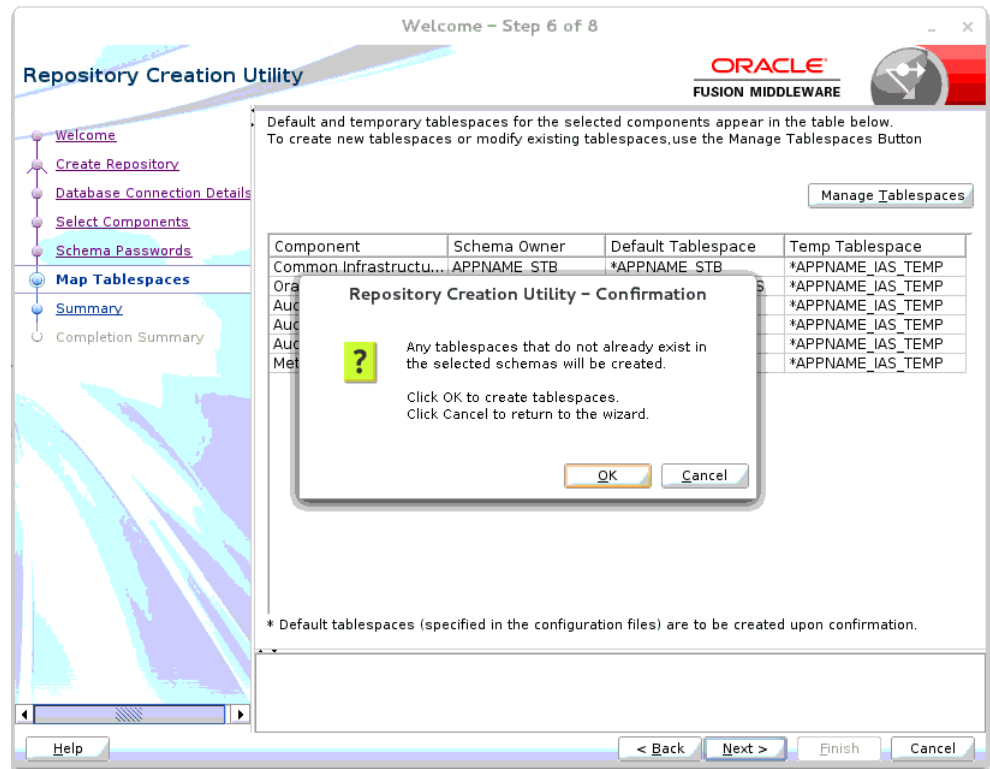
Note: This password is needed at the time of ADF domain creation.



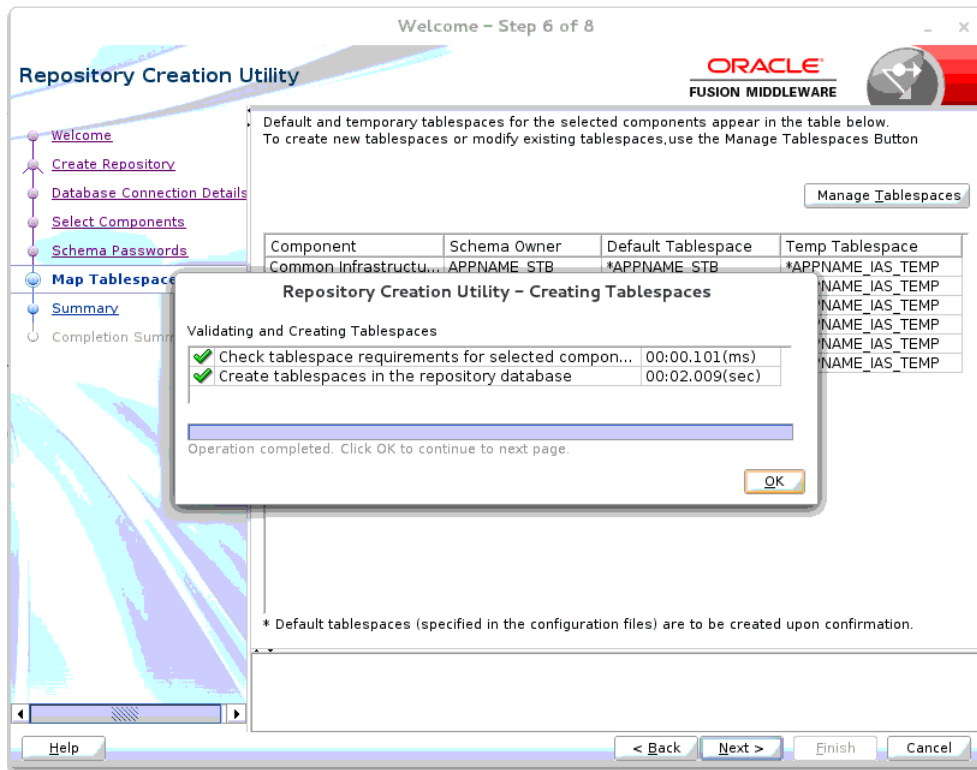
11. Provide the password and click Next.



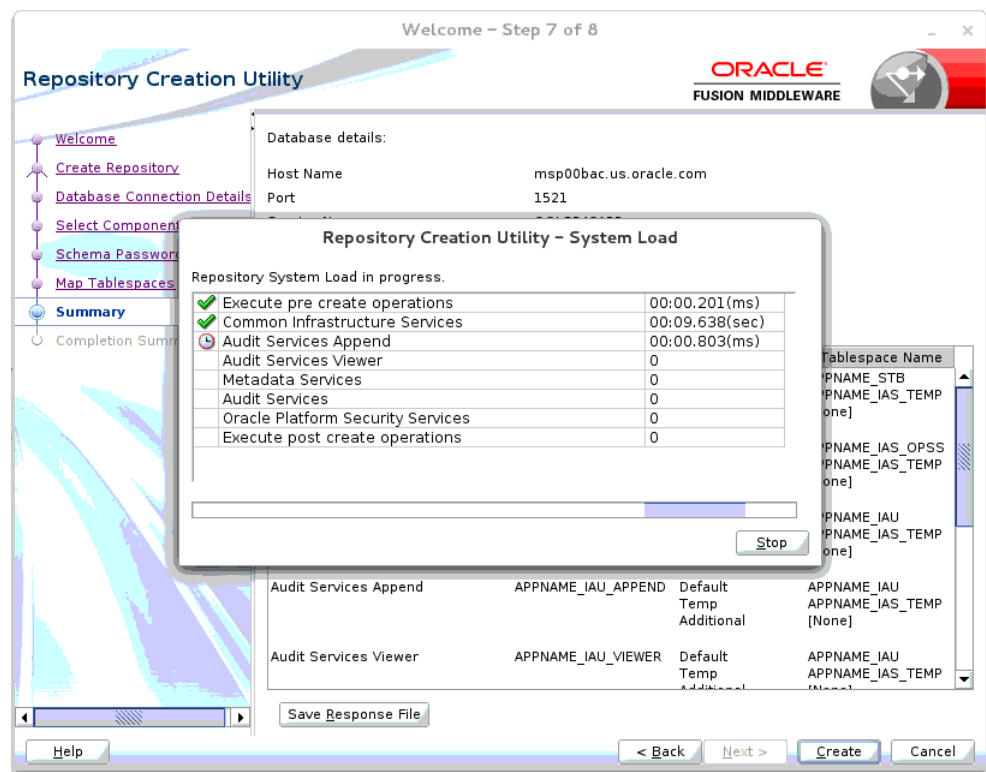
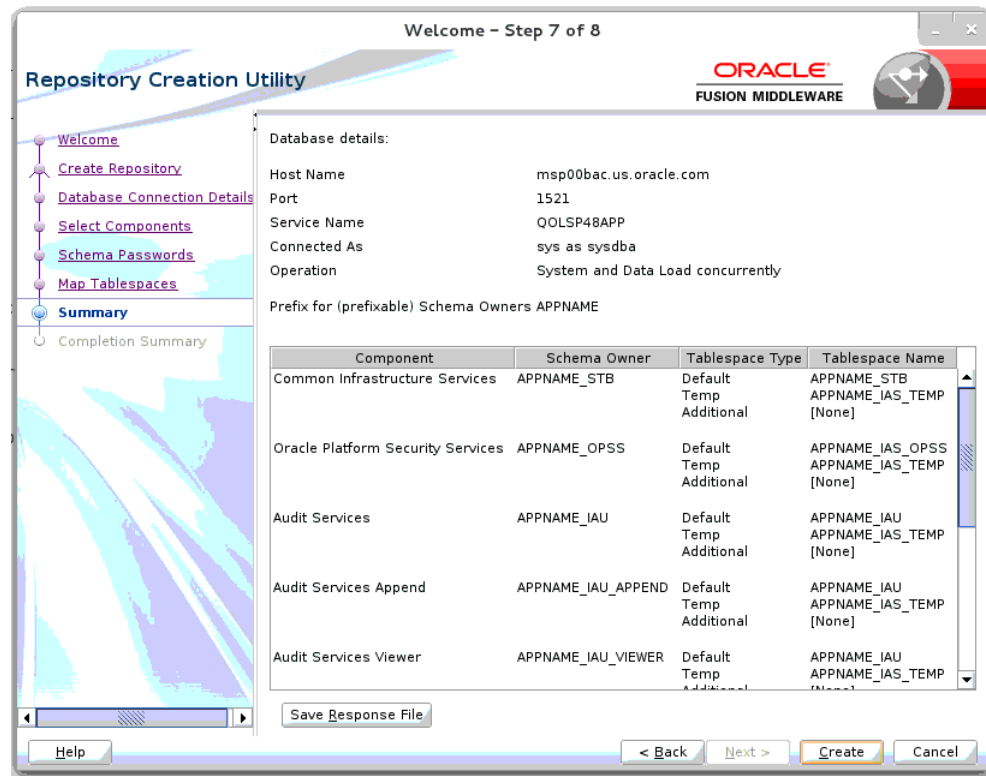
12. Click Next. A Repository Creation notification will appear. Click OK.



13. Tablespaces are created, and the progress will be displayed in a pop-up notification. When the operation is completed, click **OK**.

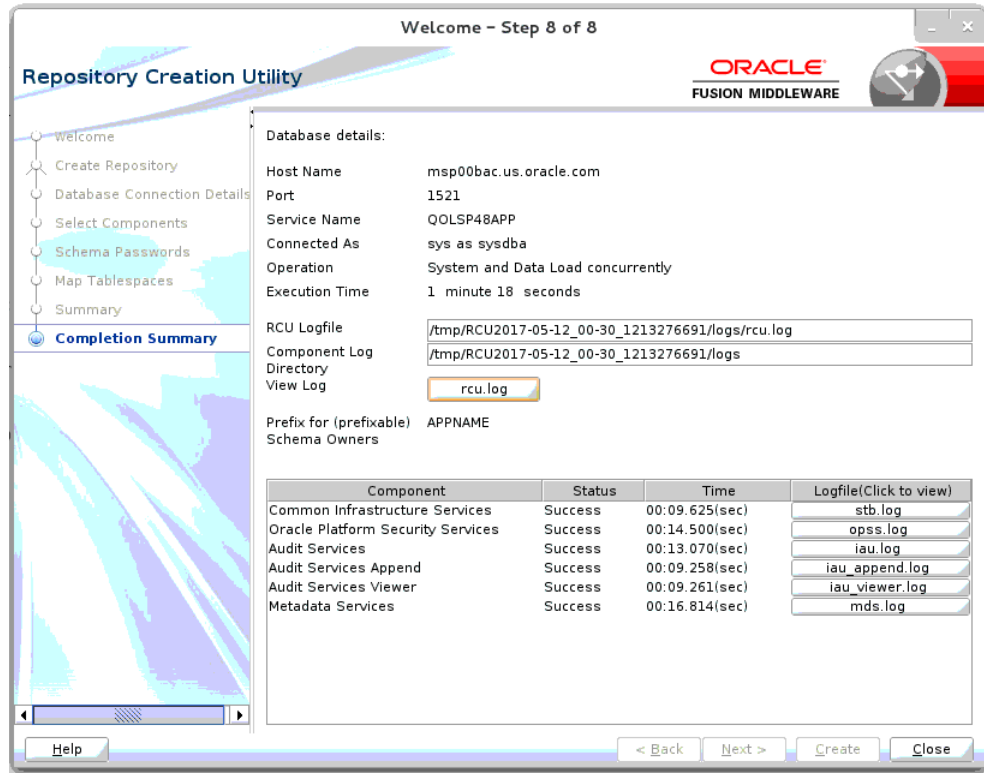


14. Click **Create**. The schema is created.



Upon successful creation of database schemas, a screen will appear with all the schemas created.

15. Click Close.



Create a New ADF Domain (with managed server and EM)

To create a new domain and managed server with ADF libraries and EM, follow the below steps:

1. Set the environment variables:

```
export JAVA_HOME=<JDK_HOME>
(Example: /u00/webadmin/products/jdk_java) [JDK_HOME is the location where
jdk has been installed)
export PATH=$JAVA_HOME/bin:$PATH
export ORACLE_HOME=<ORACLE_HOME>/
(Example: /u00/webadmin/products/wls_retail/)

cd $ORACLE_HOME/oracle_common/common/bin
(ORACLE_HOME is the location where Weblogic has been installed.)
```

2. Run the following command:

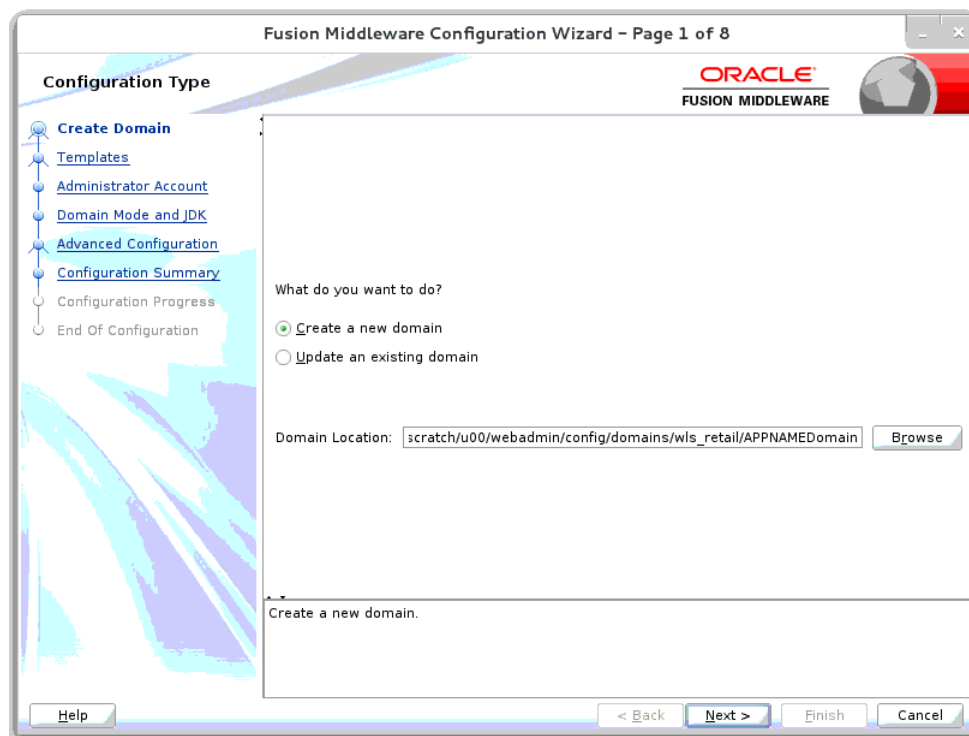
```
./config.sh
```

3. Select **Create a new domain**.

Domain location: Specify the path to the <DOMAIN_HOME>

Example: /u00/webadmin/config/domains/wls_retail/APPNAMEDomain

Click **Next**.



4. Select **Create Domain Using Product Templates**.

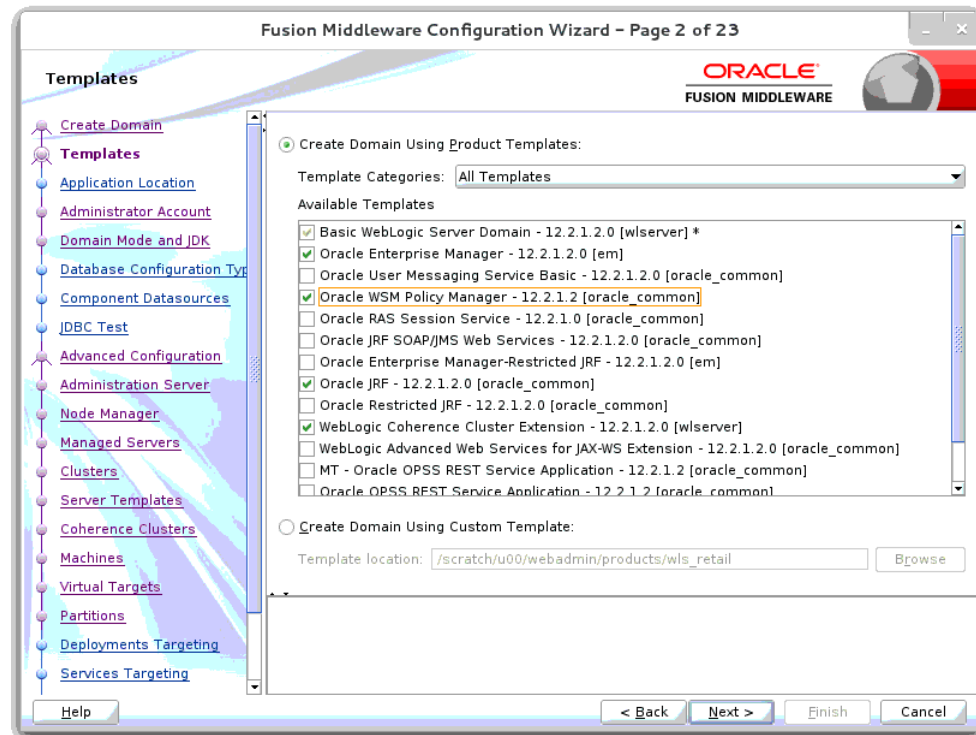
- Check the following components:
Oracle Enterprise Manager
Oracle WSM Policy Manager

Note: When Oracle Enterprise Manager component is selected, the following dependent components are selected automatically:

Oracle JRF

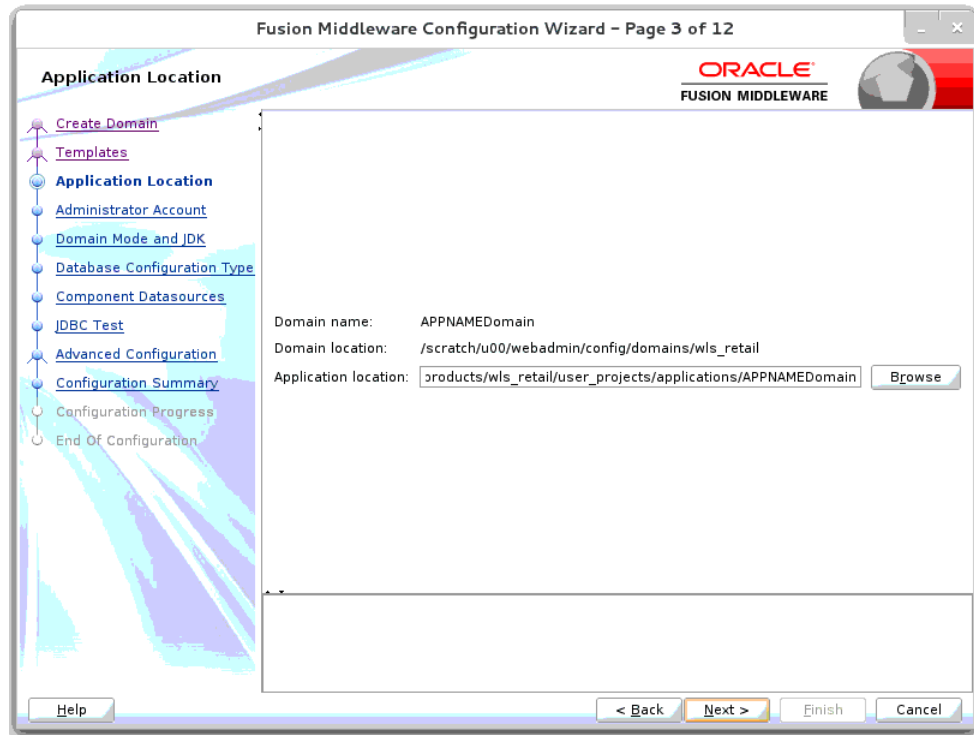
Weblogic Coherence Cluster Extension

- Click Next.



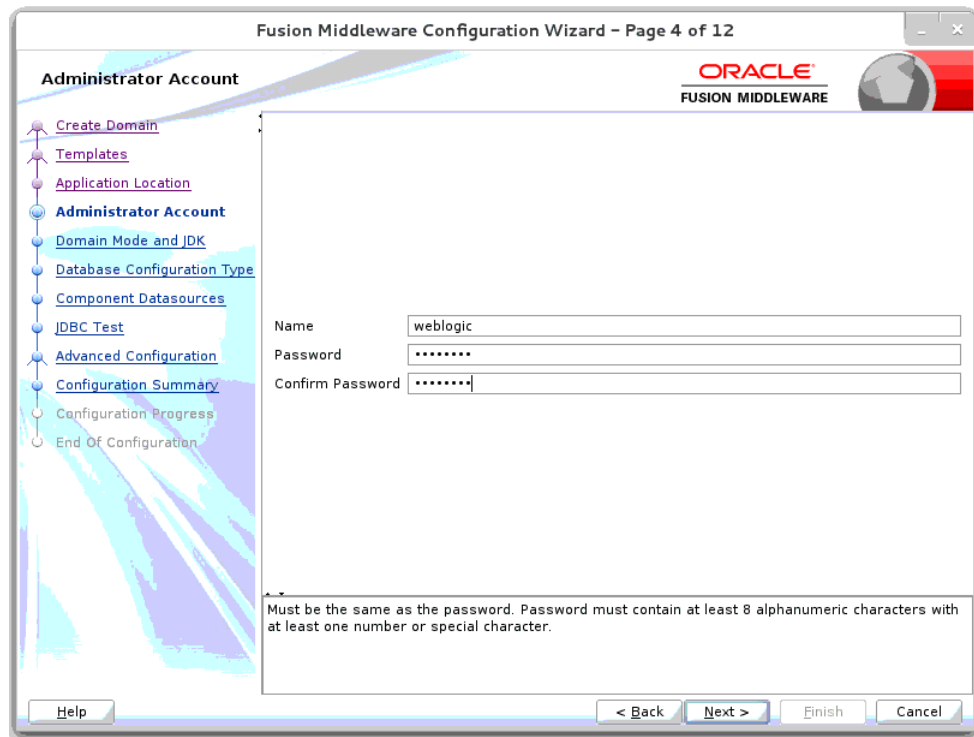
Application location: Application directory location. Example:
/u00/webadmin/config/applications/wls_retail/APPNAMEDomain

- Click Next.

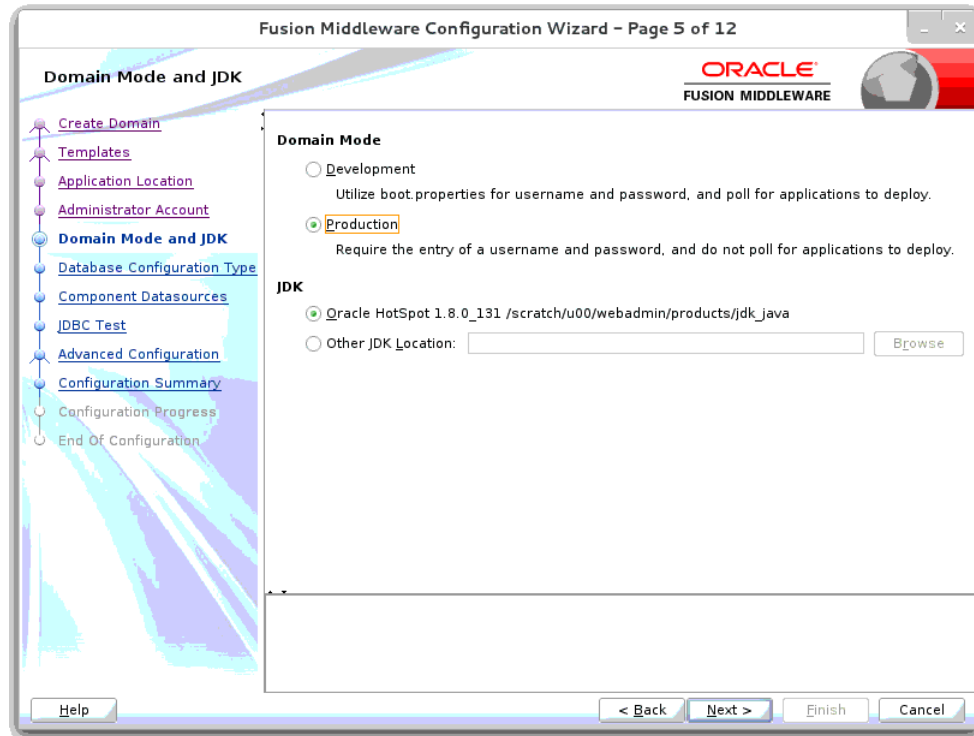


8. Provide the WebLogic administrator credentials and click Next:

- Username: weblogic
- Password: <Password>

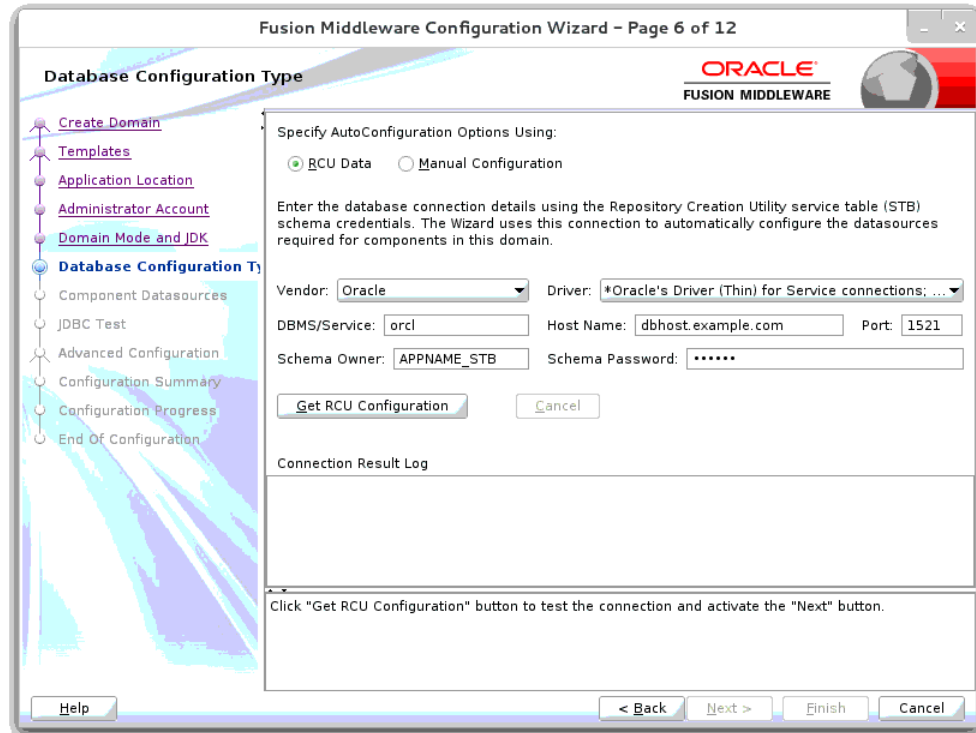


9. Select Domain Mode as Production and the JDK to use (as applicable) and click Next.

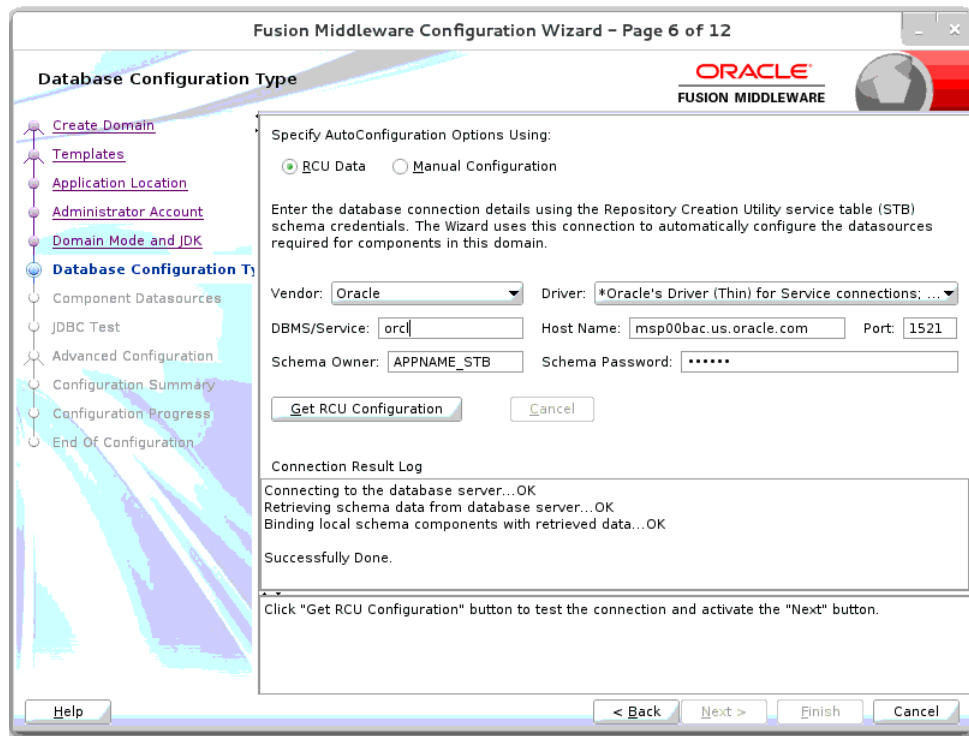


10. Select RCU Data.

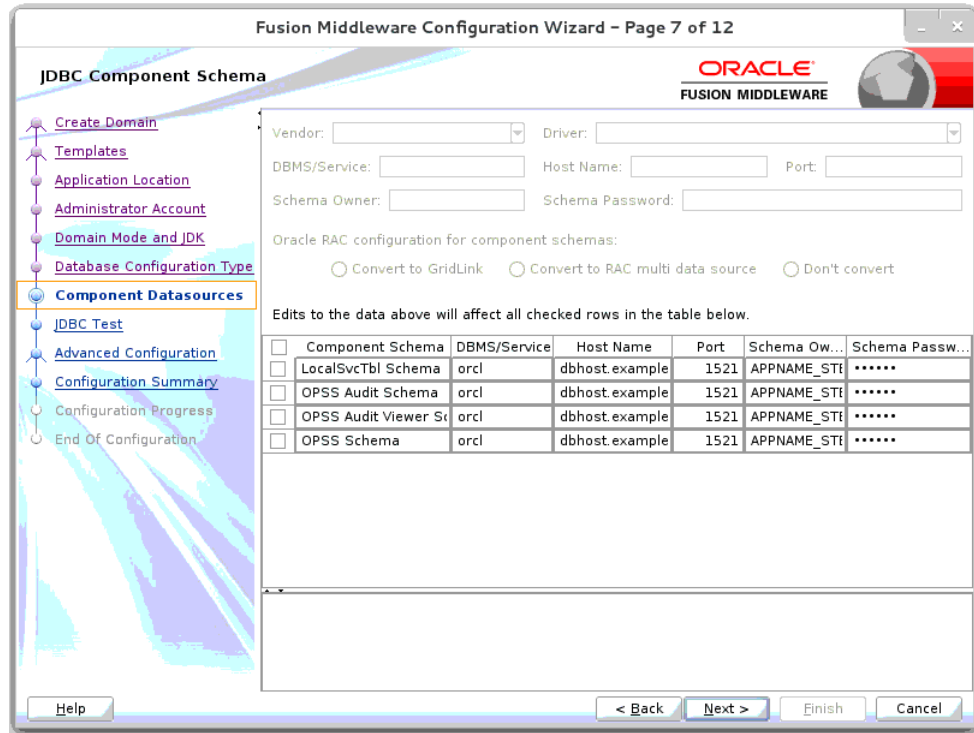
- Vendor: Oracle
- DBMS/Service: dbservicename
- Host Name: dbhostname.us.oracle.com
- Port: 1521
- Schema Owner: APPNAME_STB (Example: ALLOC_STB, ReSA_STB, etc)
- Password: <Password>. This password which was used for RCU schema creation.



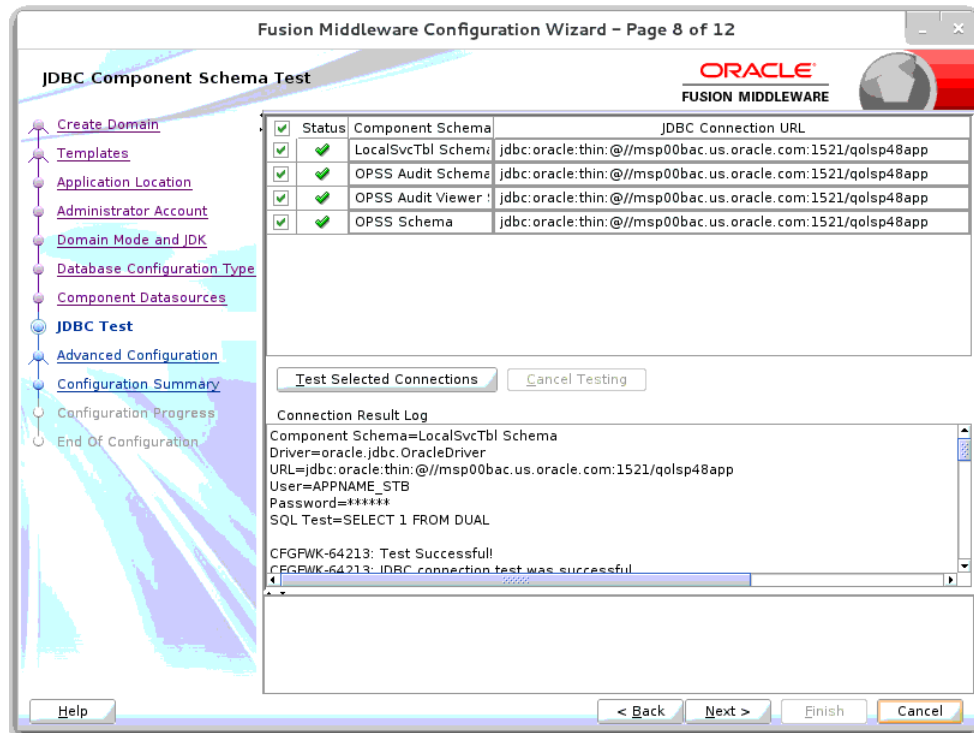
11. Click the **Get RCU Configuration** button.



12. Click **Next**.

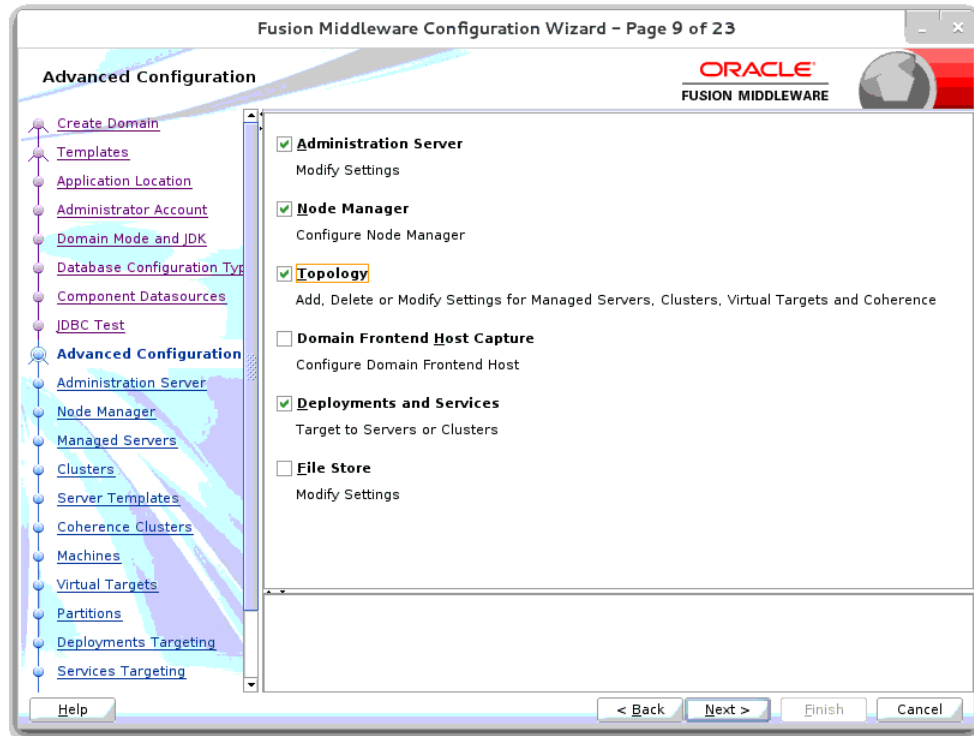


13. Click Next. The datasource connections are tested.

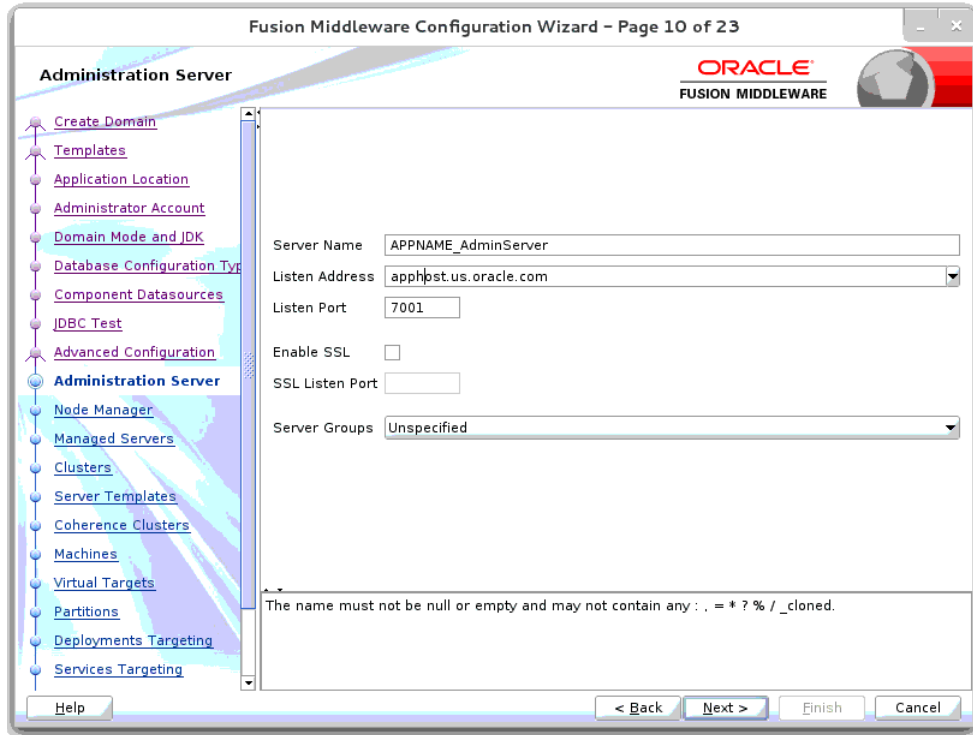


14. Click Next to continue

15. Select advanced configuration for:
- Administration Server
 - Node manager
 - Managed Servers, Clusters and Coherence
 - Deployments and Services

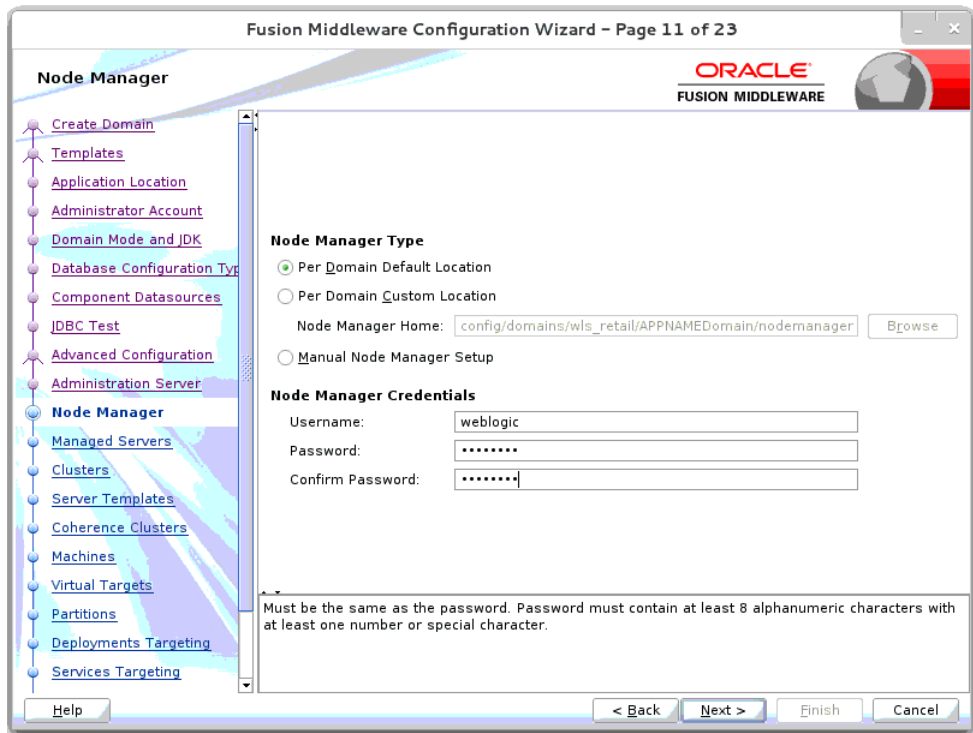


16. Configure the Administration Server:
- Server Name: <APP name>_AdminServer
 - Listen address: Appserver Hostname or IPAddress of the Appserver Host.
 - Listen port: <Port for Admin Server> Note: The port that is not already used.
 - Server Groups: Unspecified



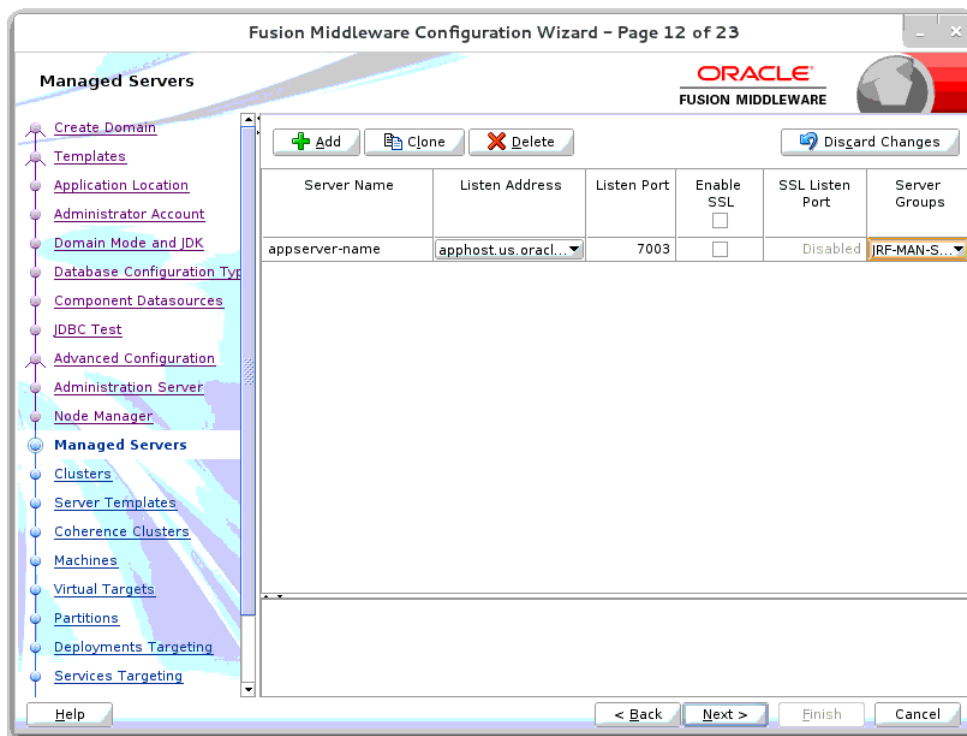
17. Configure Node Manager:

- Node manager type: Per domain default location
- Username: weblogic
- Password: <Password for weblogic>

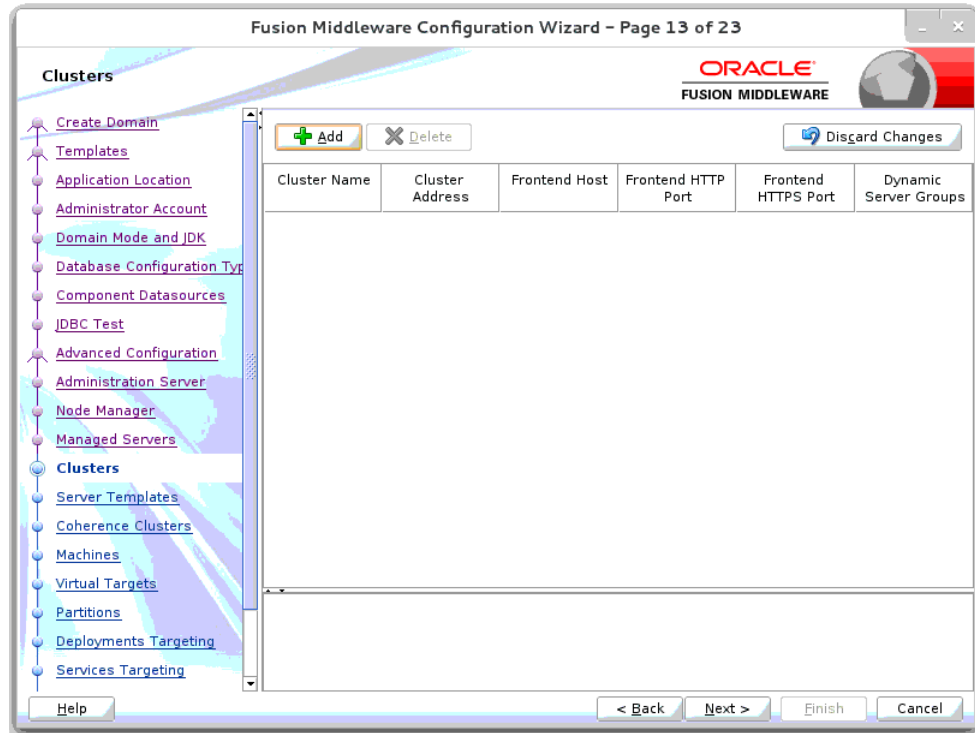


18. Click the **Add** button.

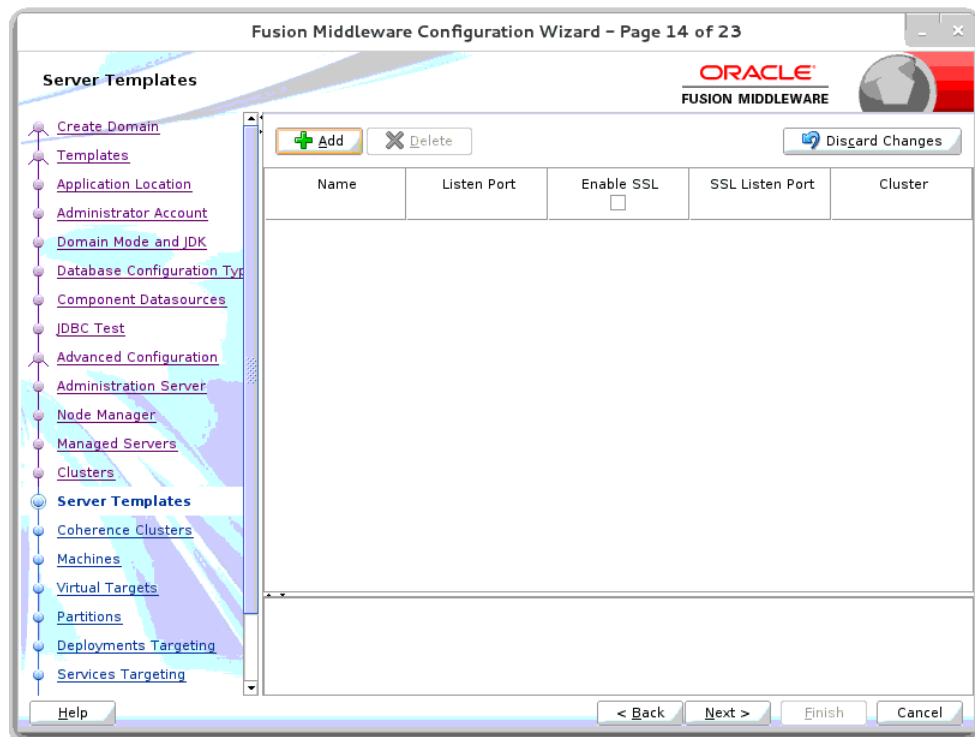
- Server Name: <appname-server>
- Listen address: Appserver Hostname or IPAddress of the Appserver Host
- Listen port: <Port for Managed Server> Note: The port used here must be a free port.
- Server Groups: JRF-MAN-SVR



19. Skip Configure Clusters and click **Next**.

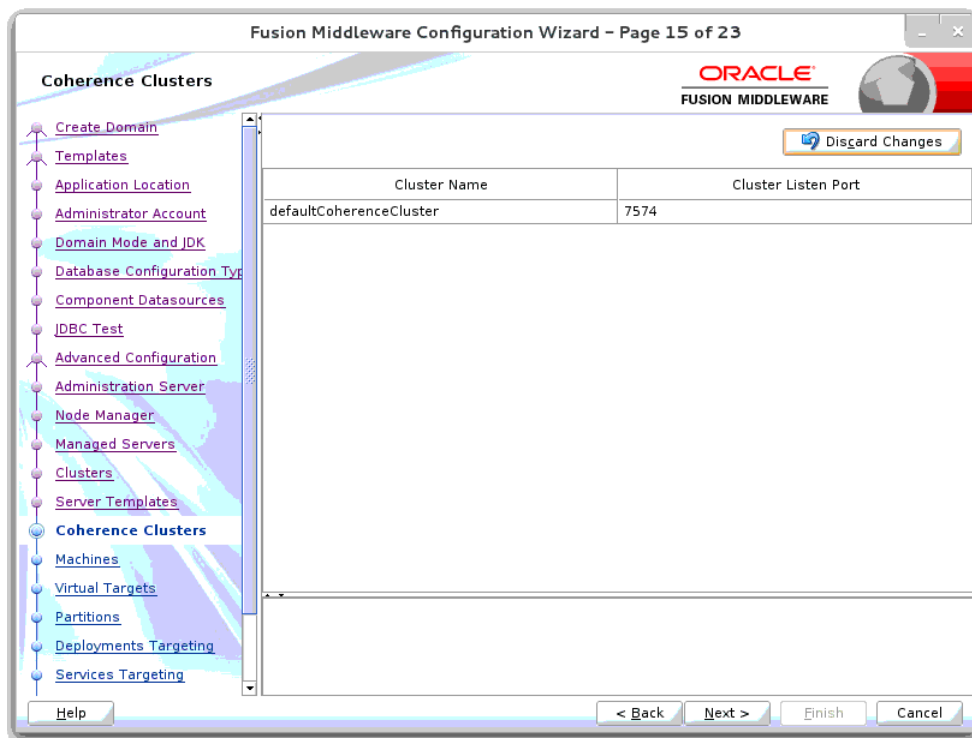


20. No change needed. Click Next.



21. Skip Server Templates and click Next.

22. Click Next.

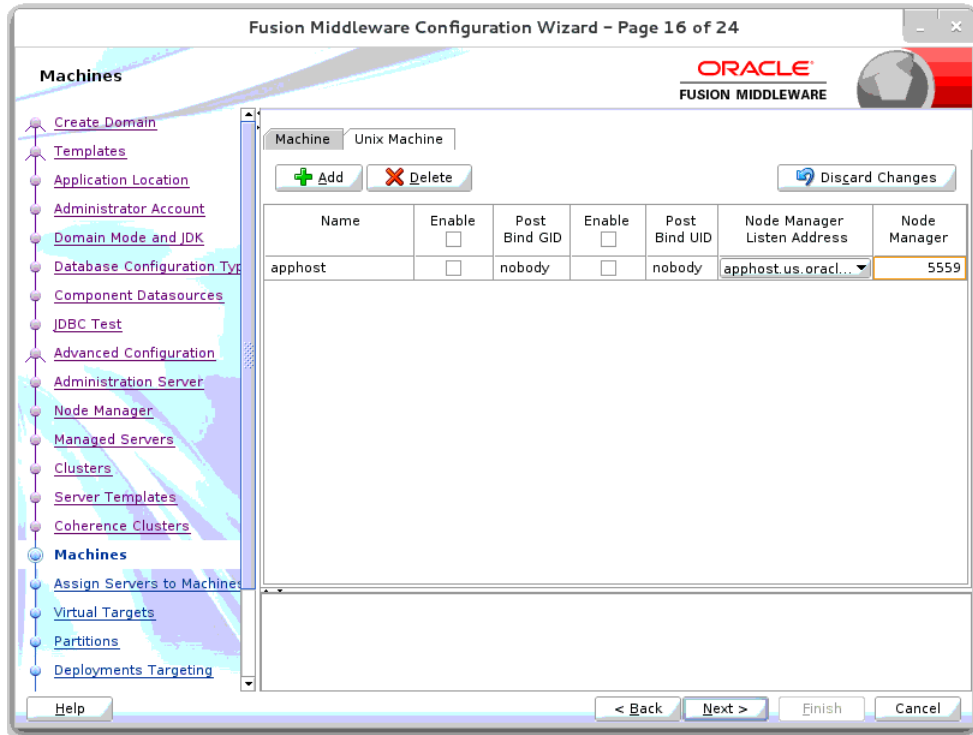


23. Configure Machines

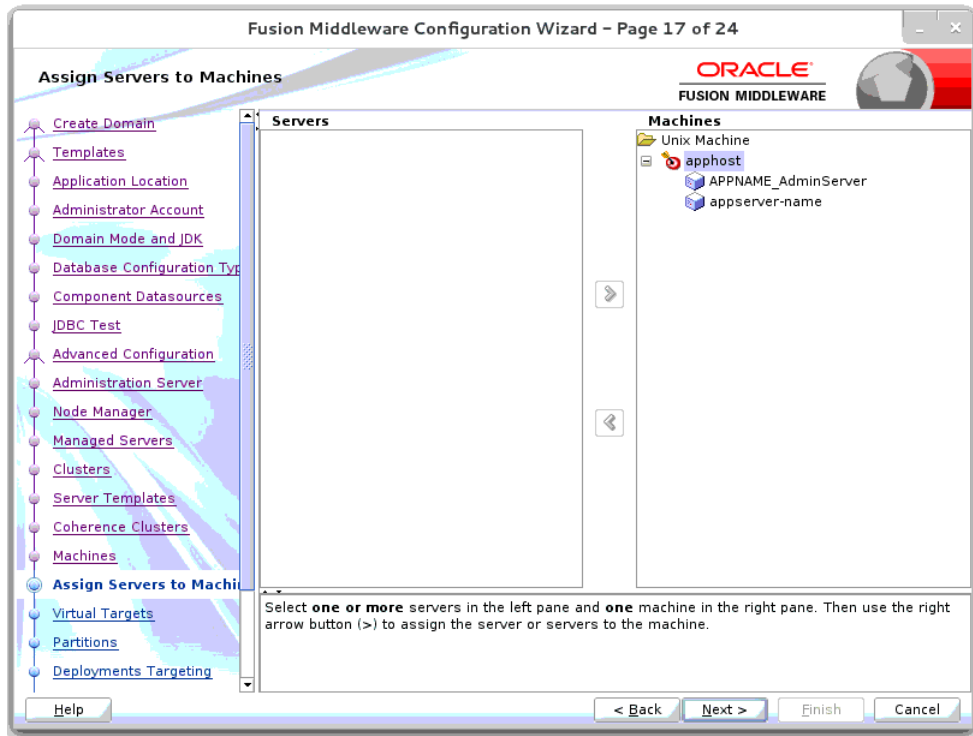
Select unix Machine :

Click the **Add** button.

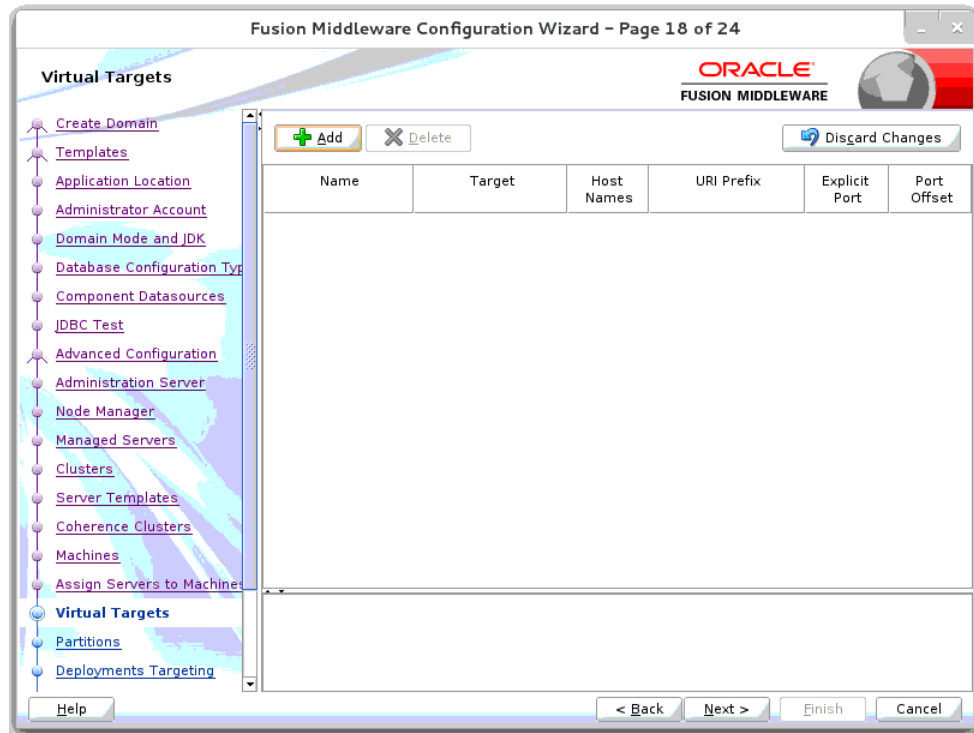
- Name: apphostname_MACHINE
- Listen address: apphostname or IPAddress
- Listen port: <Port for node manager> Note: The port used here must be a free port.



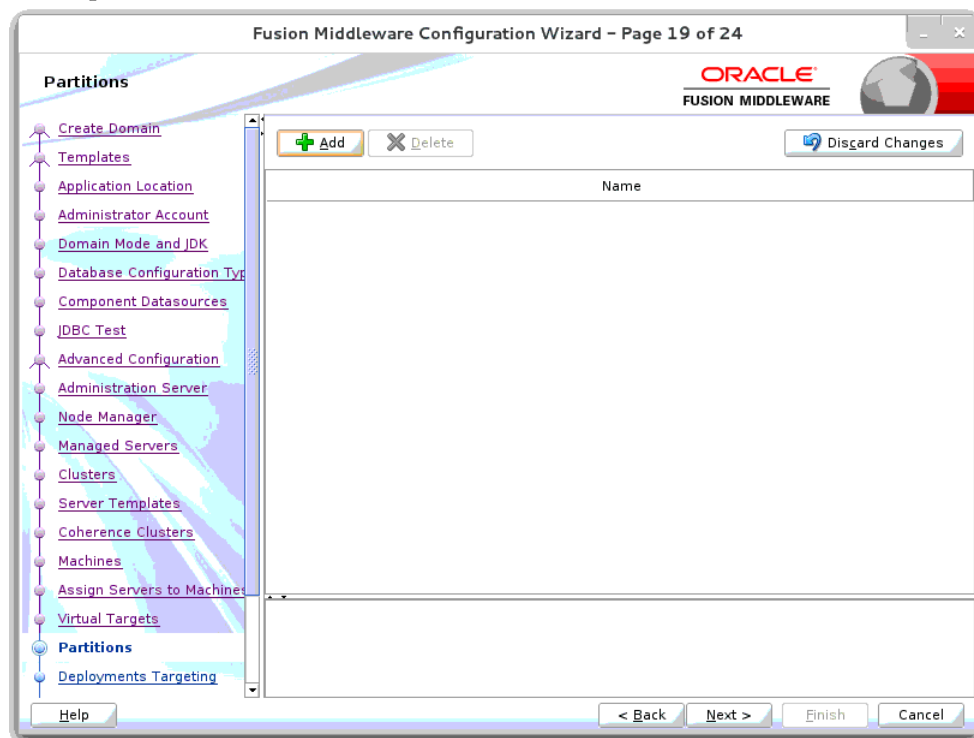
24. Assign the configured Admin server and managed servers to the new machine.



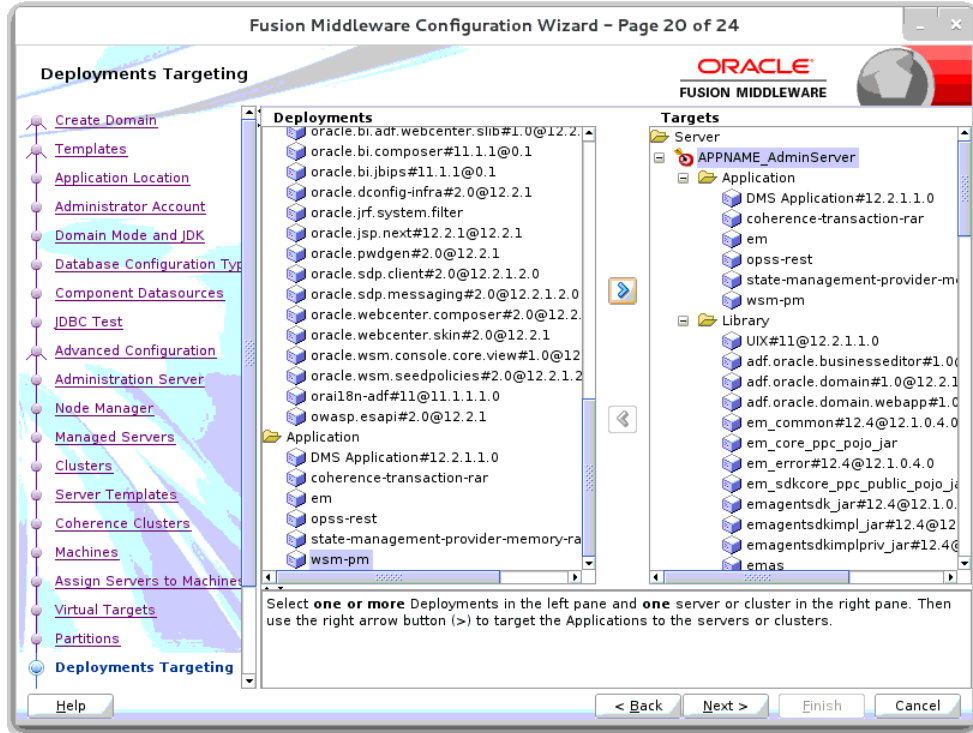
25. Skip Virtual Targets. Click Next.



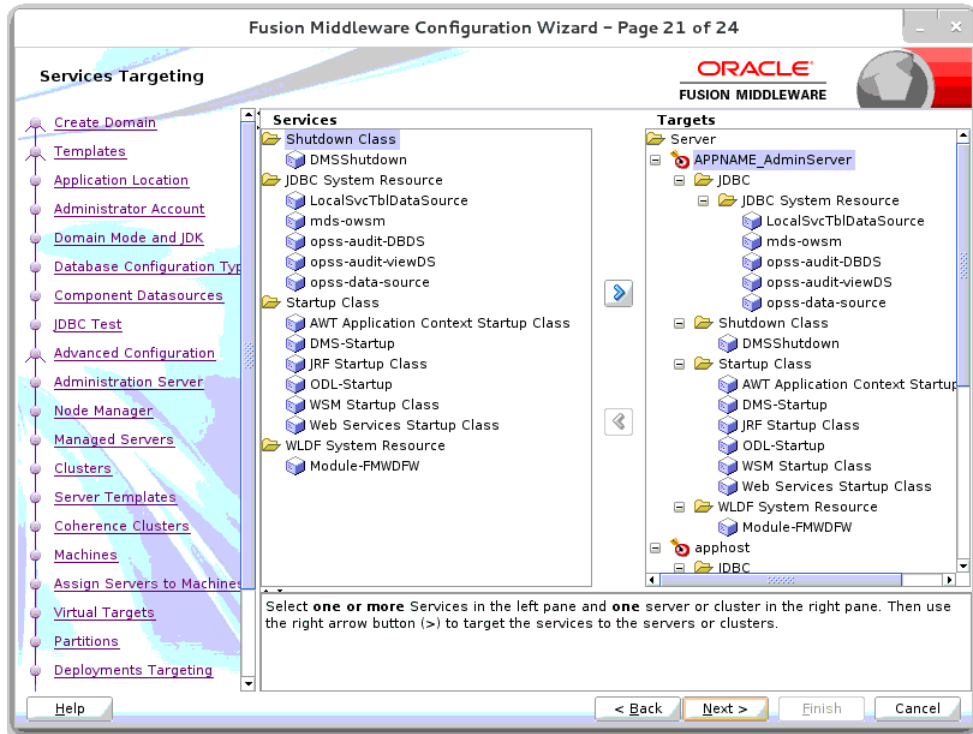
26. Skip Partitions. Click Next.



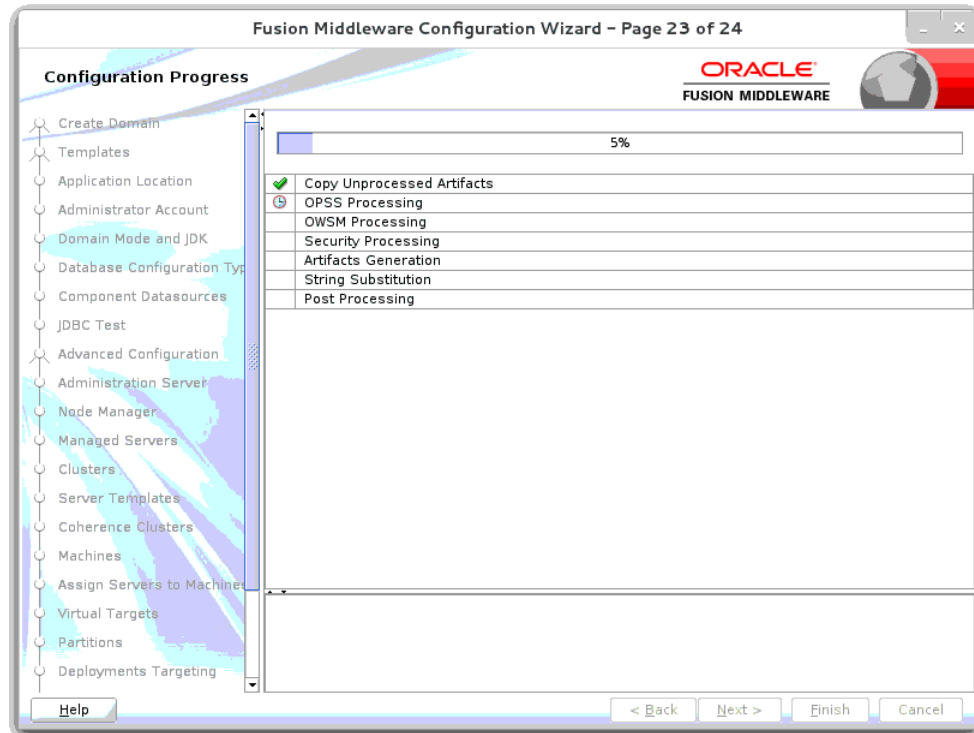
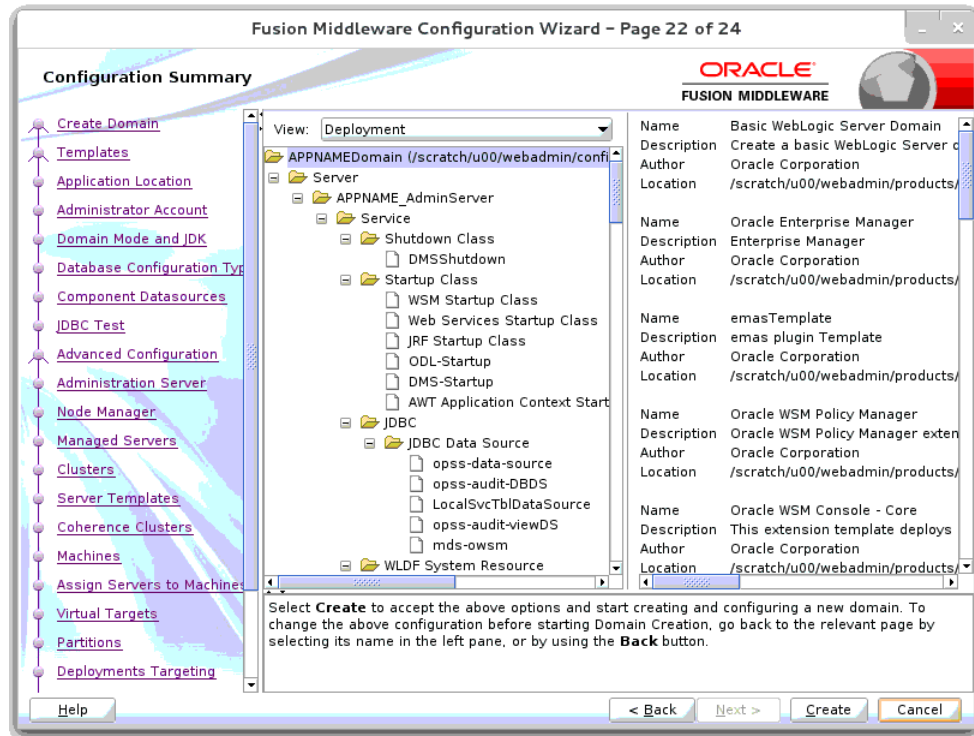
27. Target the "wsm-pm" deployment to APPNAME_AdminServer:



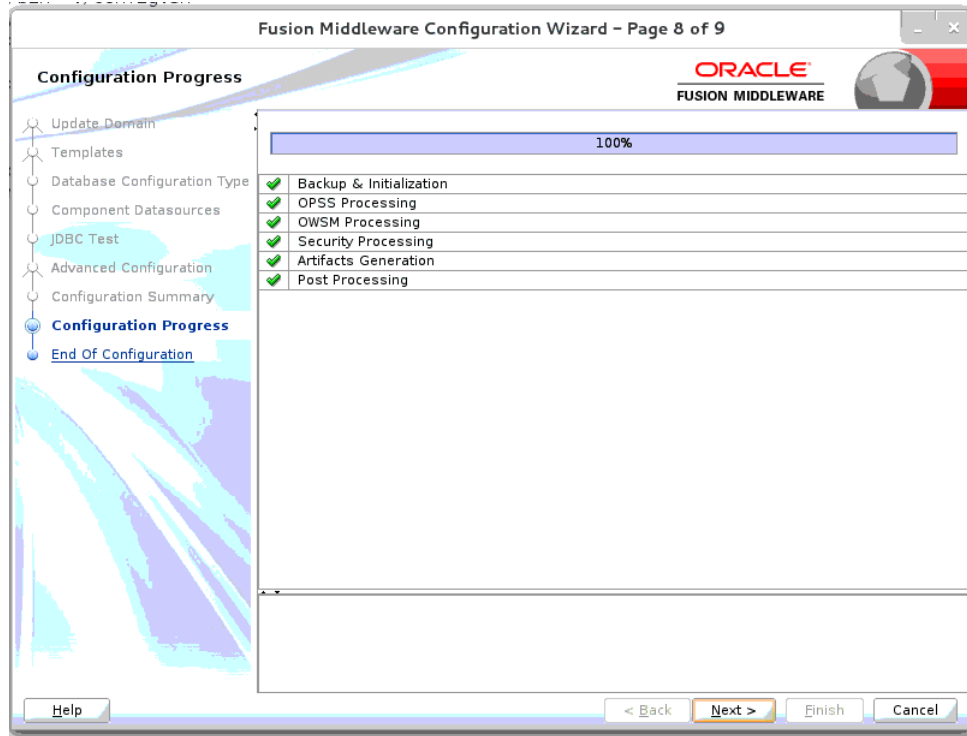
28. Click Next.



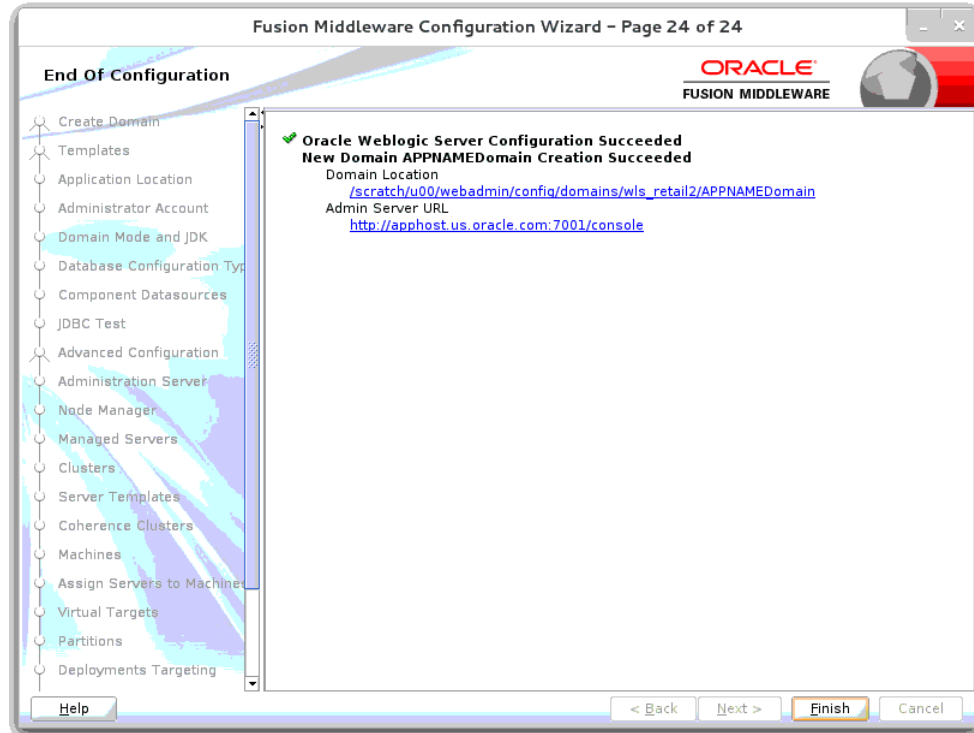
29. Click Create.



30. Click Next.



31. When the process completes, click Finish.



Start the Node Manager

1. Start the nodemanager from <DOMAIN_HOME>/bin using the following script:
nohup ./startNodeManager.sh &

Start the AdminServer (admin console)

1. Configure boot.properties for starting the Weblogic domain without prompting to username and password using the following command:
2. Create security folder at <DOMAIN_HOME>/servers/<AdminServer>/ and create boot.properties file under <DOMAIN_HOME>/servers/<AdminServer>/security
The file 'boot.properties' should have the following:

```
-----  
username=weblogic  
password=<password>  
-----
```

In the above, the password value is the password of WebLogic domain which is given at the time of domain creation.

Save the boot.properties file and start WebLogic server.

3. Start the WebLogic Domain (Admin Server) from <DOMAIN_HOME> using the following:

```
nohup ./startWebLogic.sh &
```

Example:

```
nohup /u00/webadmin/config/domains/wls_retail/RWMSdomain/  
startWebLogic.sh &
```

4. Access the Weblogic Admin console

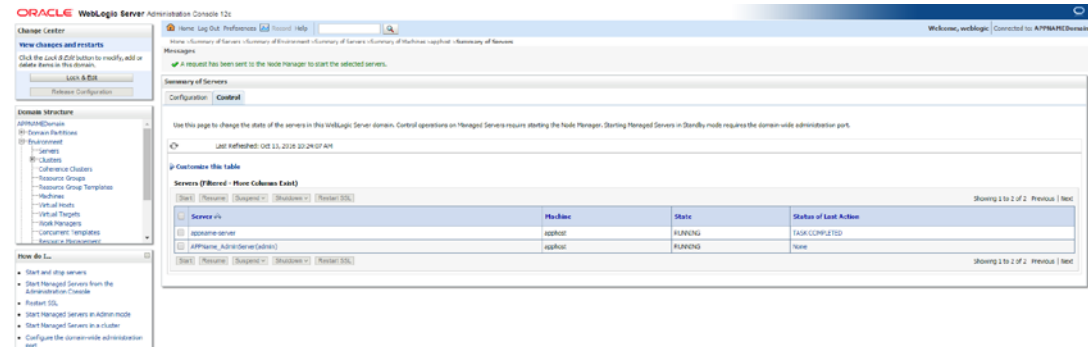
Example: http://<HOST_NAME>:<ADMIN_PORT>/console

In the below screen, provide username=weblogic and password=<weblogic password>



Start the Managed Server

After NodeManager is started, the managed servers can be started via the admin console. Navigate to Environments -> Servers and click the Control tab. Select `rwms-server` and click **Start**.



Managed Server should be up and running before configuring further steps

Run the RWMS Installer

1. This installer configures and deploys the RWMS Enhanced Navigation and Dashboards application. If you are using an X server such as Exceed, set the `DISPLAY` environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset `DISPLAY` for text mode.
2. Verify that the managed server to which RWMS Enhanced Navigation and Dashboards will be installed is currently running.
3. Change directories to `STAGING_DIR/rwms/installer`.
4. Set and export the following environment variables.

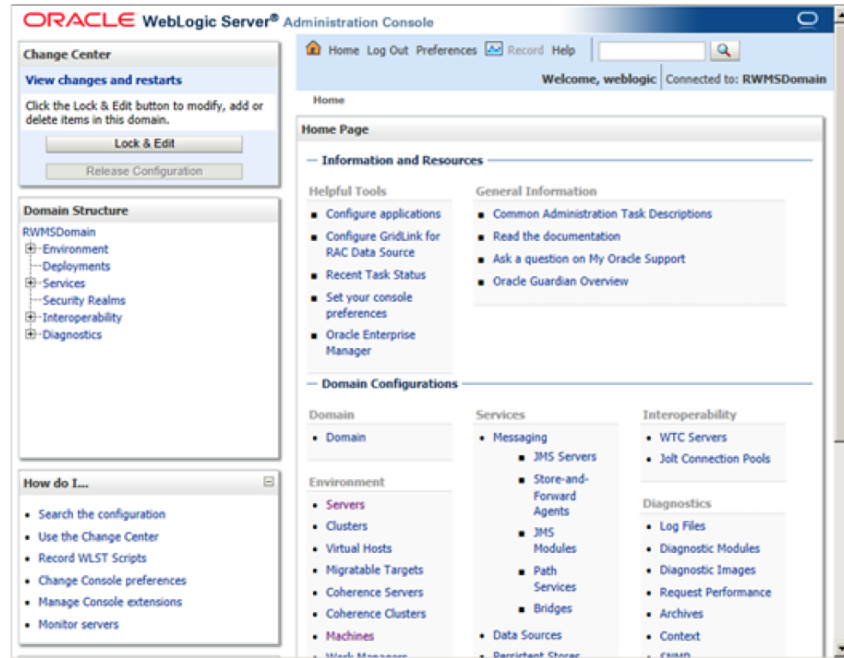
Variable	Description	Example
<code>J2EE_ORACLE_HOME</code>	The location of the WebLogic installation that contains the RWMS Enhanced Navigation and Dashboards Domain.	<code>J2EE_ORACLE_HOME=/u00/webadmin/product/wls_retail</code> <code>export J2EE_ORACLE_HOME</code>
<code>J2EE_DOMAIN_HOME</code>	The location of the WebLogic domain that contains the RWMS Enhanced Navigation and Dashboards managed server.	<code>J2EE_DOMAIN_HOME=/u00/webadmin/product/wls_retail/user_projects/domains/RWMS Domain</code> <code>export J2EE_DOMAIN_HOME</code>
<code>DISPLAY</code>	Address and display number of X server on desktop system of user running install. Optional for Enhanced Navigation and Dashboards installation	<code>DISPLAY=<IP address>:0</code> <code>export DISPLAY</code>
<code>NLS_LANG</code>	Locale setting for Oracle database client	<code>NLS_LANG=AMERICAN_AMERICA.AL32UTF8</code> <code>export NLS_LANG</code>
<code>JAVA_HOME</code>	<code>JAVA_HOME</code> should point to your valid JDK installation.	<code>JAVA_HOME=/vol.rtk/java/os_platform/jdk1.8.64bit</code>

5. Run the install.sh script. This launches the installer. After installation is completed, a detailed installation log file is created:
<STAGING_DIR>/rwms/installer/logs/rwms-install.<timestamp>.log.

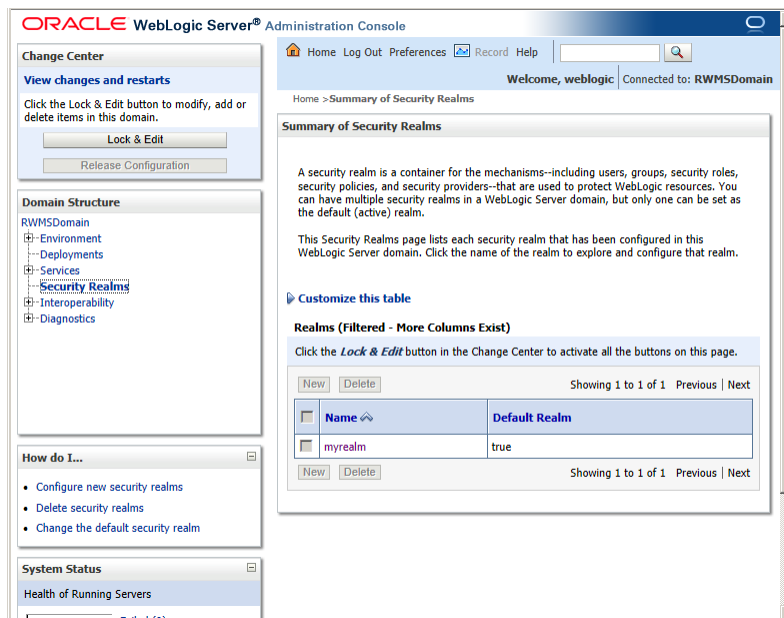
Note: See [Appendix: RWMS Enhanced Navigation & Dashboards Installation Screens](#) for details on the Enhanced Navigation & Dashboards screens and fields in the installer.

Custom Authentication Provider set up

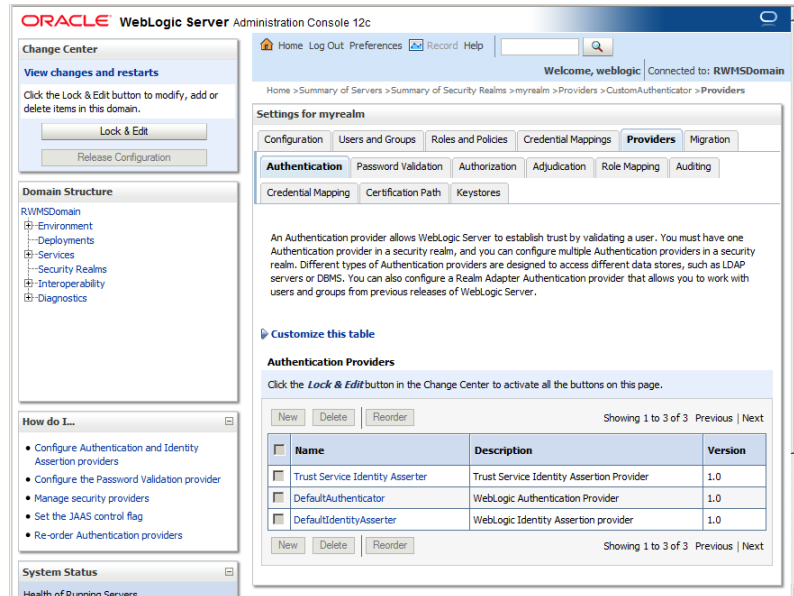
1. In the WebLogic domain - Shut down the admin and managed servers.
2. Copy the oracle.retail.rwms.security.weblogic.jar present in < STAGING_DIR >/rwms/installer/rwms15/Enhanced_Navigation_and_Dashboards/Clientlib to the WEBLOGIC_DOMAIN_HOME/lib.



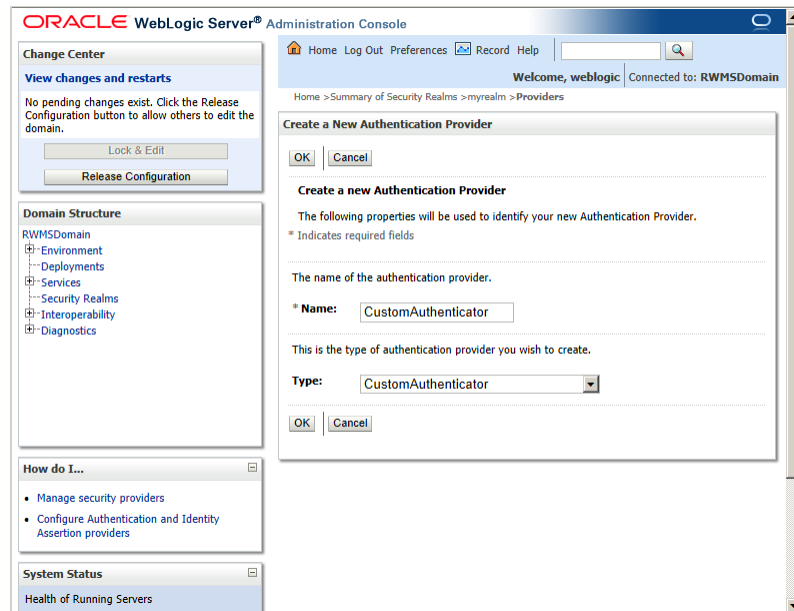
3. Start the admin server for the WebLogic domain.
4. Log into the WebLogic admin console.
5. In the Domain Structure on left hand side of screen click the Security Realms link.



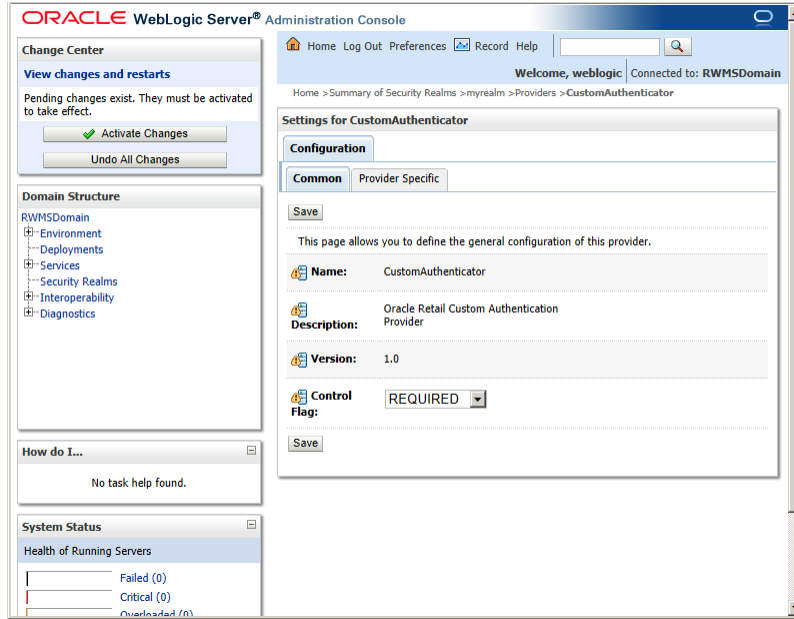
6. Click on myrealm (default realm).
7. Click on the Providers tab.



8. Click the **Lock & Edit** button.
9. Click **New**.
10. Set the provider name (example: CustomAuthenticator).
11. Select the provider type from the list: CustomAuthenticator.

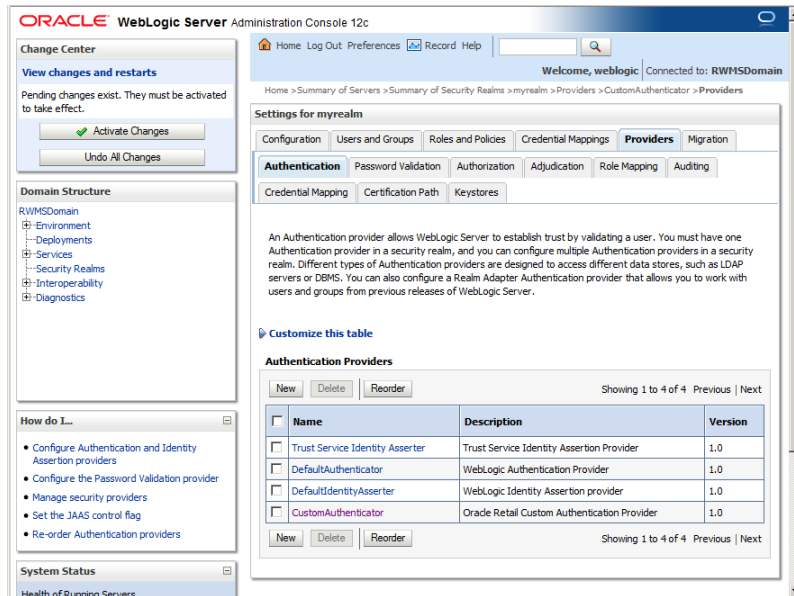


12. Click **OK**.
13. Click on the newly configured provider (CustomAuthenticator) and set the Control Flag to **REQUIRED**.



14. Click **Save**.

15. Click **Providers** link. (Top of screen – under **Welcome, <Admin_USER>**)



16. Click the **DefaultAuthenticator** link.

17. Set the default authenticator to **SUFFICIENT**.

Settings for DefaultAuthenticator

Configuration Performance Migration

Common Provider Specific

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

Save

This page displays basic information about this WebLogic Authentication provider. You can also provider is used in the login sequence.

Name: DefaultAuthenticator

Description: WebLogic Authentication Provider

Version: 1.0

Control Flag: SUFFICIENT

Save

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

18. Click **Save**.

19. Click **Providers** link. (Top of screen – under Welcome, <Admin_USER>).

ORACLE WebLogic Server Administration Console 12c

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: RWMSDomain

Home > Summary of Servers > Summary of Security Realms > myrealm > Providers > CustomAuthenticator > Providers

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing

Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

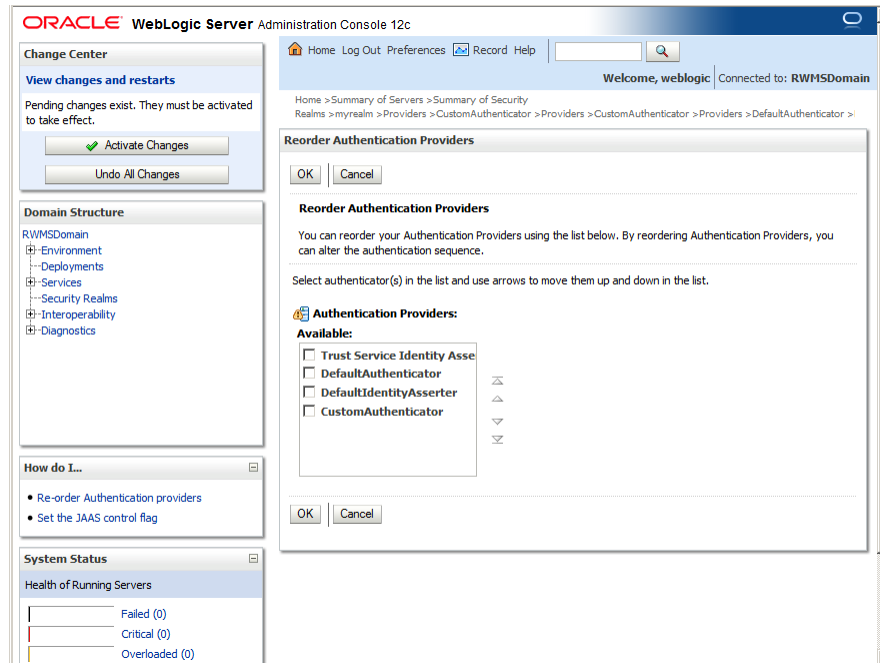
New Delete Reorder Showing 1 to 4 of 4 Previous | Next

<input type="checkbox"/>	Name	Description	Version
<input type="checkbox"/>	Trust Service Identity Asserter	Trust Service Identity Assertion Provider	1.0
<input type="checkbox"/>	DefaultAuthenticator	WebLogic Authentication Provider	1.0
<input type="checkbox"/>	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0
<input type="checkbox"/>	CustomAuthenticator	Oracle Retail Custom Authentication Provider	1.0

New Delete Reorder Showing 1 to 4 of 4 Previous | Next

20. Click Reorder button and then reorder the Authentication Providers as follows:

- Trust Service Identity Asserter
- Default Authenticator
- Default Identity Asserter
- Custom Authenticator. Re-order them accordingly.



21. Click OK.

22. Click the **Activate Changes** button and restart the entire domain.

Disable User Lock Out Settings

1. Login into the Domain.
2. Go to Security Realms and then to myrealm. Under the Configuration tab go to User Lockout.
3. Take Lock & Edit and disable the Lockout Enabled check box.
4. Click Save.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

General RDBMS Security Store **User Lockout** Performance

Save

Password guessing is a common type of security attack. In this type of attack, a hacker attempts to log in to a computer using various combinations of usernames and passwords. W security realm.

Lockout Enabled

Lockout Threshold: 5

Lockout Duration: 30

Lockout Reset Duration: 5

Lockout Cache Size: 5

Lockout GC Threshold: 400

Save

5. Activate the changes and Restart the domain.

Application: Logging File Settings

For getting clear logs of the application, the managed server's logging.xml file should be modified by adding the tags outlined below. The location of the file is as follows:

<WEBLOGIC_DOMAIN_HOME>/config/fmwconfig/servers/<RWMS Managed Server>

Under <log_handlers> tag (Replace <RWMS Managed Server> with your managed server name):

```
<log_handler name='rwmsapp-handler'
class='oracle.core.ojdl.logging.ODLHandlerFactory'
filter='oracle.dfw.incident.IncidentDetectionLogFilter'>
  <property name='path' value='${domain.home}/servers/<RWMS Managed
Server>/logs/rwmsapp-diagnostic.log' />
  <property name='maxFileSize' value='10485760' />
  <property name='maxLogSize' value='104857600' />
  <property name='encoding' value='UTF-8' />
  <property name='useThreadName' value='true' />
  <property name='supplementalAttributes'
value='J2EE_APP.name,J2EE_MODULE.name,WEBSERVICE.name,WEBSERVICE_PORT.name,composi
te_instance_id,component_instance_id,composite_name,component_name' />
</log_handler>
```

Under `<loggers>` tag:

Note: you may want to adjust the logging level of FINEST to reduce the amount of log messages):

```
<logger name="oracle.retail.rwms" level="FINEST">  
  <handler name="rwmsapp-handler"/>  
</logger>
```


Manual Installation and Configuration Tasks

This section covers the installation and configuration of external components and other dependencies for RWMS 15.0.2.

Prerequisites

The following prerequisites have been assumed to have been completed successfully:

- The RWMS Database Schema installation into a running Oracle 12cR1 database platform must be completed.
- The RWMS application components must be installed into a working WebLogic 12.12.1.2 domain.
- The RWMS Enhanced Navigation and Dashboard must be installed into a working WebLogic 12.2.1.2 domain

Path References Used in this Section

The directory structures outlined in this section will vary according to your specific configuration of the target WebLogic domain. We will therefore use the labels outlined in the following table as references to directories used for the configuration of external dependencies for RWMS 15.0.2 throughout the rest of this section.

Note the directory names for each of the labels outlined below for your installation environment.

Label	Description	Example Directory
<STAGING_DIR>	The application installation staging directory as defined for the source installation files	
<INSTALLATION_NAME>	The installation name of the RWMS system as defined for the application installation	rwms15inst
<WLS_JAVA_HOME>	The JDK location used by the WebLogic Server instance being installed to.	/u00/webadmin/jdk_java
<BI_REPOSITORY>	The BI Publisher 11g reports repository where the reports are present.	/u00/webadmin/product/fmw/wls_obiee/user_projects/domains/bifoundation_domain/config/bi_publisher/repository
<RWMS_REPOSITORY>	The RWMS User report templates repository.	<BI_REPOSITORY>/Users/Guest/RWMS REPOSITORY
<RWMS_SYSTEM_REPOSITORY>	The RWMS System reports repository	<BI_REPOSITORY>/Users/Guest/RWMS SYSTEM REPOSITORY

Label	Description	Example Directory
<WLS_SERVER_CONFIG_DIR>	The configuration base directory for the WebLogic server instance supporting the Oracle Forms environment.	/u00/webadmin/product/10.3.6/WLS_Forms/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS
<FORMS_INSTALL_BASE>	The Oracle Forms base configuration directory.	<DOMAIN_HOME>/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1.2/config
<RETAIL_HOME>	The final installation directory for RWMS application to run from (such as forms, extras, and reports).	/u00/webadmin/rwms15inst/

Note: In the following sections, the Oracle BI EE 12c installation steps are sample. Refer to the *Oracle Business Intelligence 11g Installation Guide* for more information.

BI Server Component Installation Tasks

Oracle BI Publisher is used as the main RMS, RWMS, REIM, and SIM reporting engine and can be used in conjunction with external printing solutions like label printing. This section describes the installation of Oracle BI Publisher as a server application within WebLogic 12.2.1.2. One deployment of BI Publisher can be used for any and all of the RMS, RWMS, REIM, and SIM reports.

When installing BI Publisher 11g, refer to the appropriate Fusion Middleware guides for the installation of the product in a WebLogic server environment.

Installation Process Overview

Installing the BI Publisher server as a standalone web application in a WebLogic server involves the following tasks:

1. Run RCU to create BIPublisher related database schemas and other db objects.
2. Install Oracle BI EE under an existing Weblogic Server (WLS) 10.3.6 and choose "Enterprise Install".
3. Configure Oracle BI EE, create default bifoundation_domain and configure component "Business Intelligence Publisher" only.
4. Select the BIPlatform schema for update of the ORACLE 12c DB
5. Configure ports and document and test the URL's that are created.

The following post-installation tasks are involved once BI Publisher has been installed:

6. Set up and copy the RWMS BI Publisher Report Templates produced for RWMS.
7. Configure the BI Publisher repository. Set security model, add users, assign roles, add reports, add printers, set repository path, set data source, etc.
8. Set up for the RWMS application specific configuration files to integrate BI Publisher with the RWMS online app.

BiPublisher 12c only - Installation Process Overview

Oracle BiPublisher must be installed in a standalone setup, it cannot be incorporated with OBIEE Analytics as this would prevent Guest access to the BiPublisher reports.

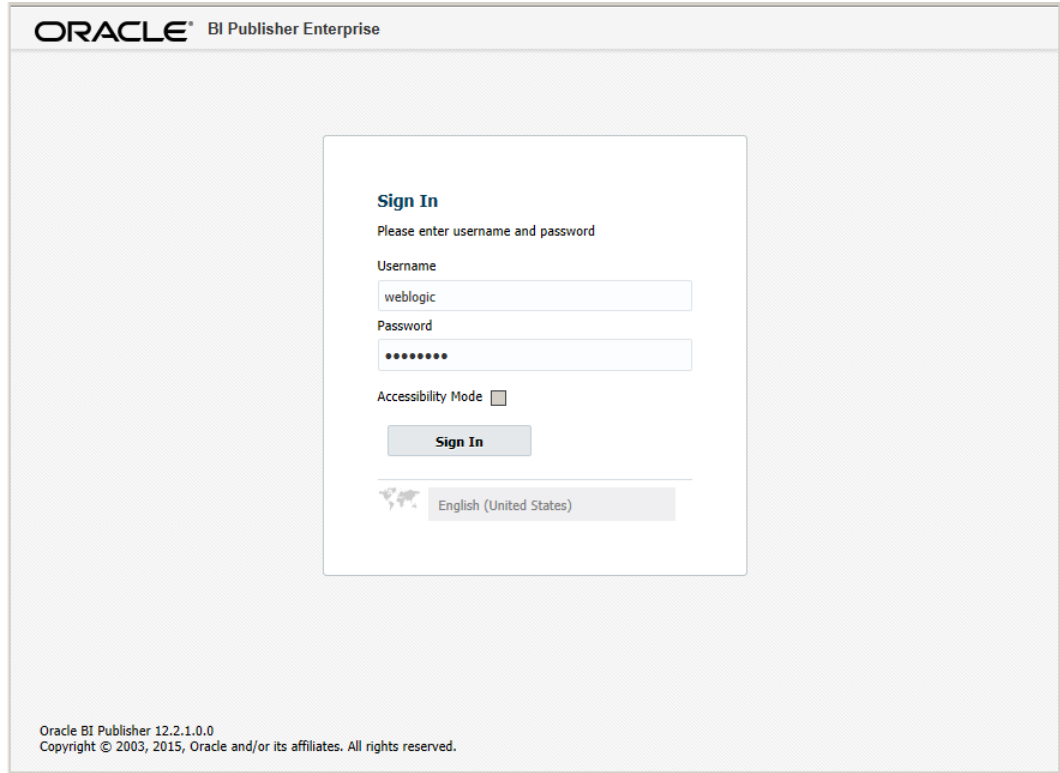
The BiPublisher install steps are documented here:

<http://docs.oracle.com/middleware/12212/bip/index.html>

Once BiPublisher is installed follow the post install steps below to configure the reports.

Post install steps for BiPublisher 12C

1. Test your BIPublisher installation, Get the xmlpserver url from your Installation Screen and launch xmlpserver. Login with the credentials you entered in your Oracle BI EE configuration (weblogic / password). Example URL:[http://\[obiee_host\]:\[obiee_server_port\]/xmlpserver](http://[obiee_host]:[obiee_server_port]/xmlpserver)

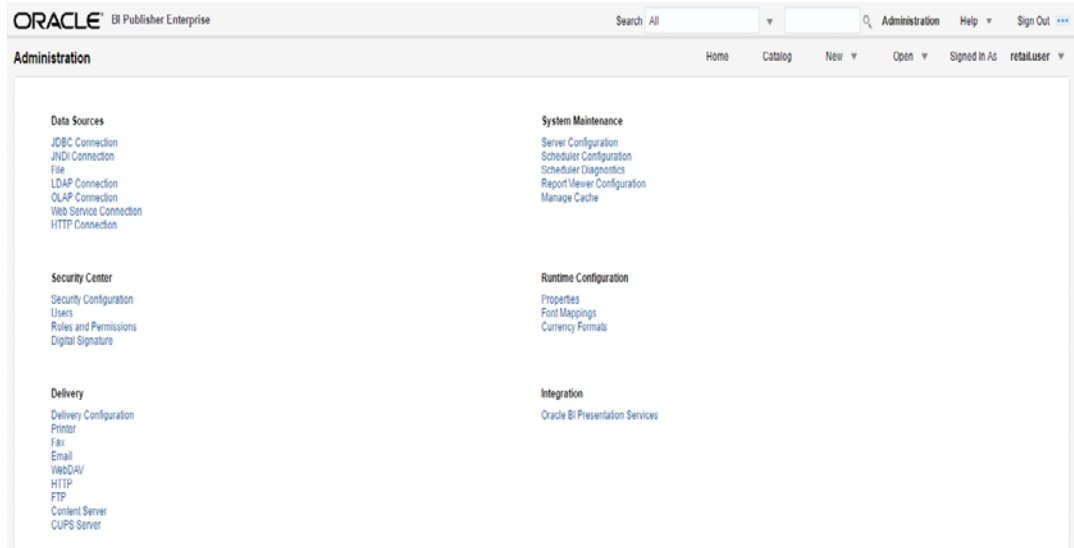


The image shows the Oracle BI Publisher Enterprise sign-in interface. At the top left, it says "ORACLE BI Publisher Enterprise". The main content is a "Sign In" box with the following elements:

- Sign In** (Section Header)
- Please enter username and password
- Username**: Input field containing "weblogic"
- Password**: Input field with masked characters "••••••"
- Accessibility Mode**: A checkbox that is currently unchecked.
- Sign In** button
- Language selection: A dropdown menu showing "English (United States)" with a globe icon.

At the bottom left of the page, the text reads: "Oracle BI Publisher 12.2.1.0.0 Copyright © 2003, 2015, Oracle and/or its affiliates. All rights reserved."

2. After sign on, select "Administration".

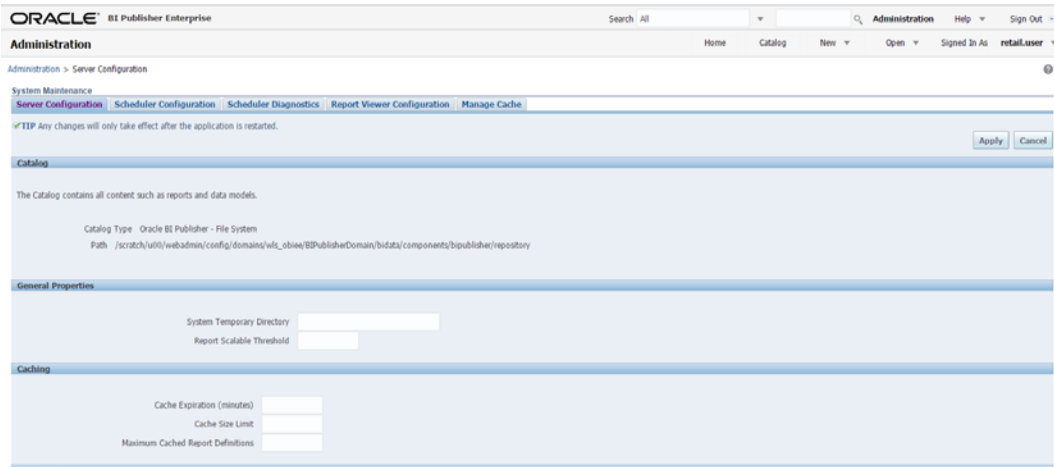


The image shows the Oracle BI Publisher Enterprise Administration page. At the top, it says "ORACLE BI Publisher Enterprise". The page has a navigation bar with "Search All" and "Administration Help Sign Out". Below the navigation bar, the page is titled "Administration" and has a sub-navigation bar with "Home Catalog New Open Signed In As retailuser".

The main content area is divided into several sections:

- Data Sources**: JDBC Connection, JNDI Connection, File, LDAP Connection, OLAP Connection, Web Service Connection, HTTP Connection
- Security Center**: Security Configuration, Users, Roles and Permissions, Digital Signature
- Delivery**: Delivery Configuration, Printer, Fax, Email, WebDAV, HTTP, Content Server, CUPS Server
- System Maintenance**: Server Configuration, Scheduler Configuration, Scheduler Diagnostics, Report Viewer Configuration, Manage Cache
- Runtime Configuration**: Properties, Font Mappings, Currency Formats
- Integration**: Oracle BI Presentation Services

3. On the System Maintenance Section, click **Server Configuration**.

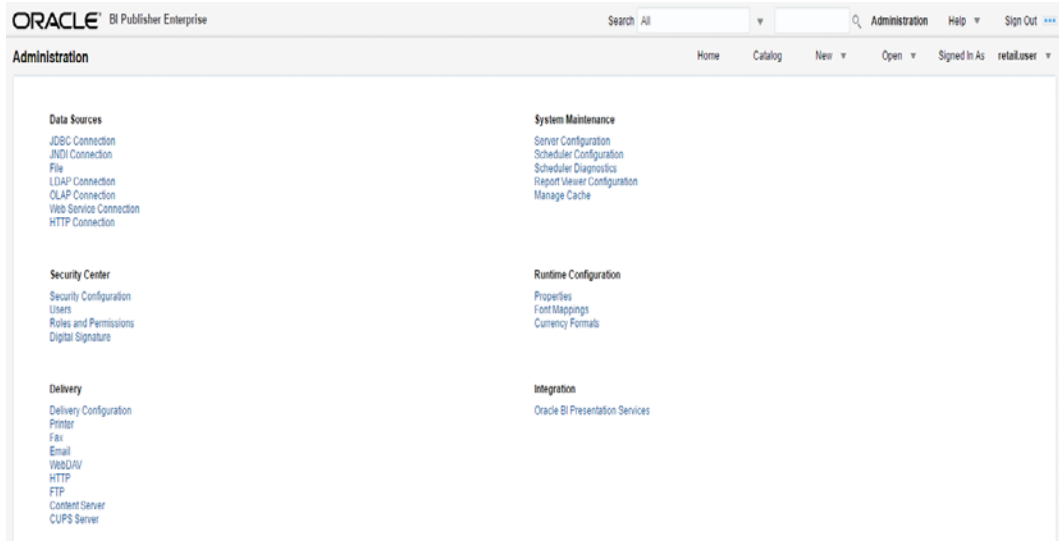


4. On this screen - In the Server Configuration Folder section, enter the path to your repository.

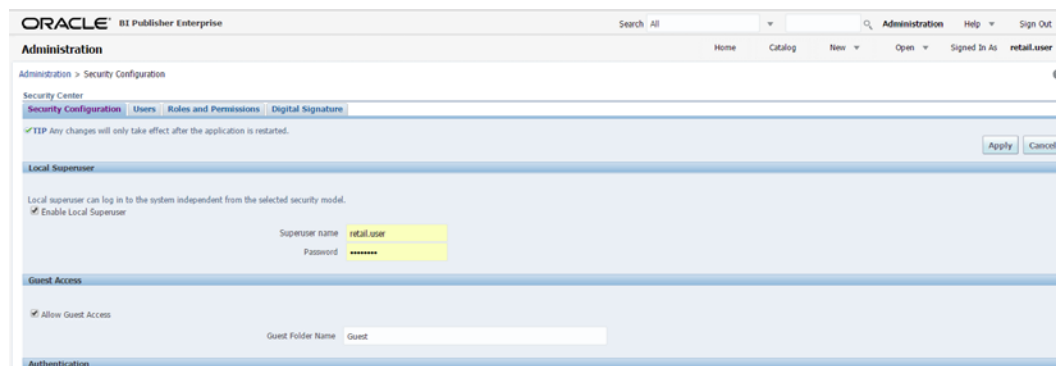
- This is the path you entered in the Configuration Section and Catalog Section:
Example: \$<OBIEE_DOMAIN_HOME>/bidata/components/bipublisher/repository

5. Click **Apply**.

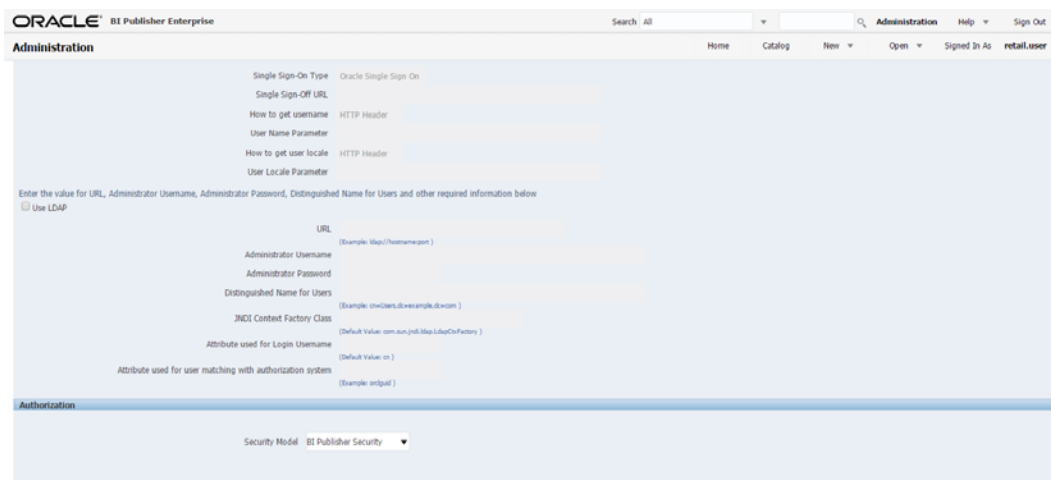
6. Click Administration link at top of screen.



- Click on the Security Configuration link under the Security Center to setup a super user and apply the BI Publisher security model.



- Enable a Superuser by checking the “Enable Local Superuser” box and by entering name and password on the corresponding fields on this screen.
- Mark “Allow Guest Access” check box. Enter “Guest” as Guest Folder Name.
- Click **Apply**.
- Scroll down the screen and locate the Authorization section:



- Select BI Publisher Security from the Security Model list.
- The default user name for the BI Publisher Security Model is Administrator.
- On the password text field, enter a value that you can remember. It is going to be the password for Login to xmlpsrver.
- Click **Apply**.
 - Leave BI Publisher up while completing the next section.

Installing the RWMS BI Publisher Templates

In this section, we have outlined how the RWMS report templates are installed into the appropriate BI server repositories. RWMS requires two sub directories to be created within the path indicated by <BI_REPOSITORY>.

Report files are placed by the installer in the directory, RETAIL_HOME/reports, and must be copied into the newly created BI server repository directory.

1. Change directory to the <RETAIL_HOME>/reports used for the application install. This directory contains a RWMS directory with "RWMS REPOSITORY" and "RWMS SYSTEM REPOSITORY" subdirectories. These two directories contain reports directories whose names reflect the names of report templates provided with RWMS.
2. Copy the RWMS directory and all of its subdirectories to <BI_REPOSITORY>/Reports/Guest. This will copy over the RWMS REPOSITORY and RWMS SYSTEM REPOSITORY directories according to the following table paths.

Example:

```
cp -R RWMS "
/u00/webadmin/product/fmw/wls_obeie/user_projects/domains/bifoundation_dom
ain/config/bipublisher/repository/Reports/Guest/ "
```

Placement of Report Templates

Directory	RWMS Repository	RWMS System Repository
asn_recv_package_audit_sys	✓	✓
asn_receiving_receipt_sys	✓	✓
best_before_date	✓	
bill_of_lading_sys	✓	✓
container_manifest_sys	✓	✓
gift_card_report_sys		✓
inventory_by_item	✓	
inventory_by_location	✓	
labor_performance_day_summary	✓	
labor_performance_detail	✓	
labor_performance_week_summary	✓	
pack_slip_labels_sys		✓
pick_package_audit_sys	✓	✓
pts_containers_to_close_sys		✓
quality_audit_sys	✓	✓
receiving_receipt_sys	✓	✓
receiving_register	✓	

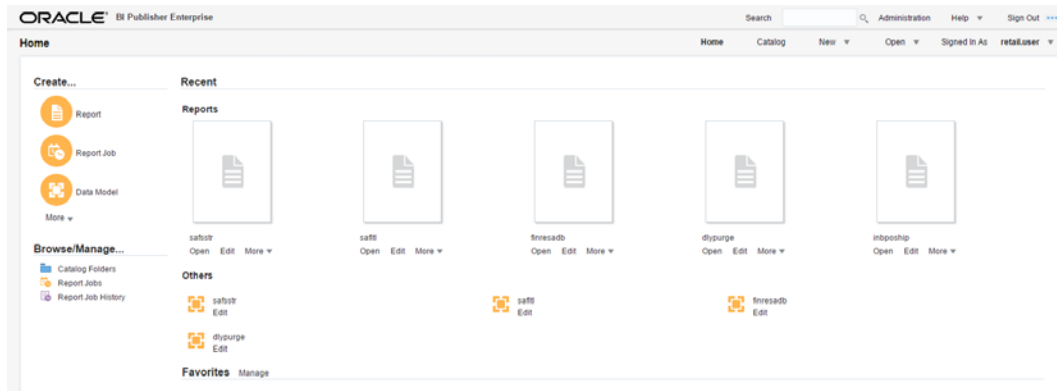
Directory	RWMS Repository	RWMS System Repository
recv_package_audit_list_sys	✓	✓
return_to_vendor	✓	
rtv_advice_sys	✓	✓
unit_pick_group_sys		✓
unresolved_appointment	✓	
error_log	✓	
cycle_count_log	✓	
confirm_fwd_case_picks_sys		✓
qualified_container_sys	✓	✓
pick_labels_lbl		✓
receiving_labels_lbl		✓
ship_labels_lbl		✓
ticketing_lbl		✓
generic_lbl		✓
prd_classification_code_sys		✓
system_control_parameter_sys	✓	✓
user_summary_sys		✓
user_task_history_sys	✓	✓

Configuring the RWMS JDBC connection

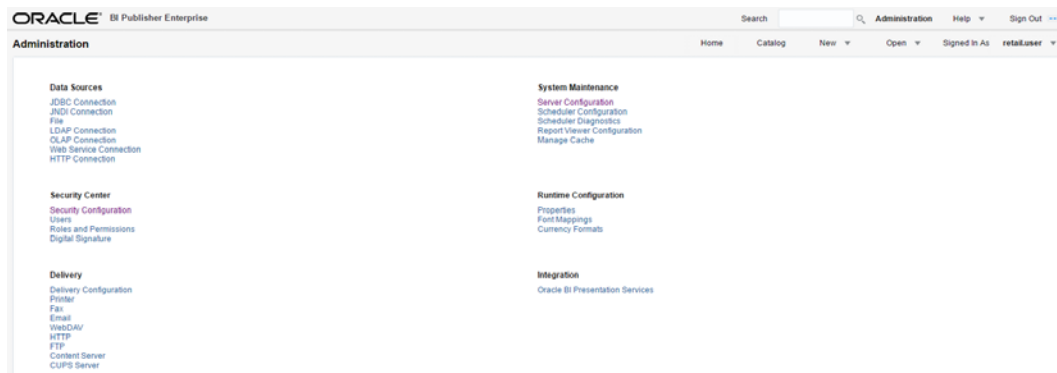
Follow the below steps to configure a JDBC connection for the RWMS Data Source, which is required for RWMS reports.

1. If not still logged into BIPublisher –
 - Login with the credentials you entered in your Oracle BI EE configuration. (weblogic / password)
2. If the server was restarted –
 - Login as the super user that was created in prior security setup steps.

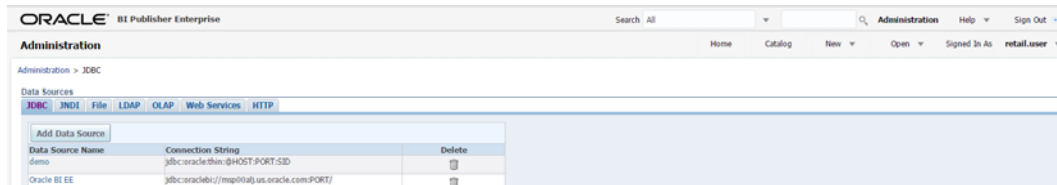
Note: You will not be able to login to xmlpserver as weblogic any more because we have already changed the Security Model.



3. Click the **Administration** link at top of screen



4. Select the JDBC Connection hyperlink in the Data Sources lists.



5. Click the **Add Data Source** button.

The screenshot shows the Oracle BI Publisher Administration console. The breadcrumb navigation is 'Administration > JDBC > Add Data Source'. The page title is 'Add Data Source'. There are 'Apply' and 'Cancel' buttons at the top right. The 'General' tab is selected. The page contains the following fields and options:

- Data Source Name:** Text input field.
- Driver Type:** Dropdown menu set to 'Oracle 12c'.
- Database Driver Class:** Text input field set to 'oracle.jdbc.OracleDriver'.
- Connection String:** Text input field with a placeholder '(Example: jdbc:oracle:thin:@[host]:[port]:[sid])' and the value 'jdbc:oracle:thin:@[host]:[port]:[sid]'.
- Use System User:** Checked checkbox.
- Username:** Text input field set to 'retail.user'.
- Password:** Password input field with masked characters.
- Pre Process Function:** Text input field.
- Post Process Function:** Text input field.
- Use Proxy Authentication:** Unchecked checkbox.
- Test Connection:** Button.

6. Enter the appropriate details for the RWMS data source. Click **Test Connection** to test the connection on the screen once the data is entered.

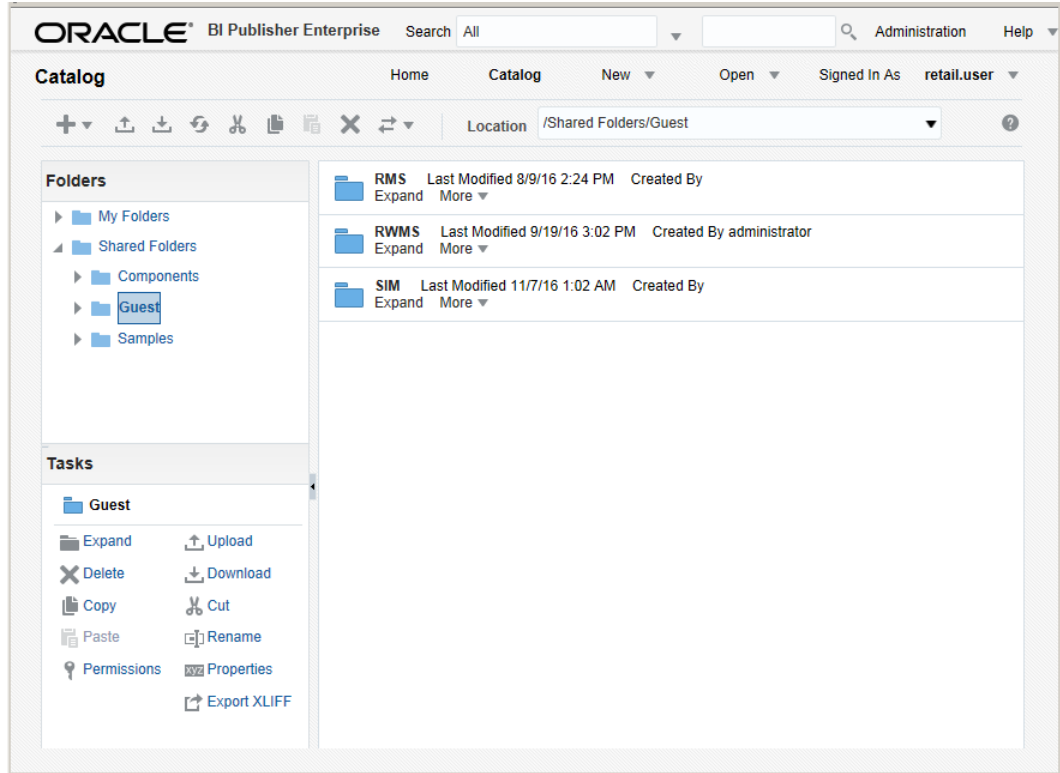
- Data Source Name: RWMS
 - Must be RWMS due to code dependencies.
- Driver type is ORACLE 12c
- Database driver class should be oracle.jdbc.OracleDriver.
- Connection string is similar to this example:
 - Pluggable: jdbc:oracle:thin:@dbhostname:1521/servicename
 - Non- Pluggable dbc:oracle:thin:@dbhostname:1521:SID
- Enter the username and password for the RWMS application user's data source. Click **Test Connection** to test the connection on the screen once the data is entered.

7. Scroll to the bottom of the screen and check the **Allow Guest Access** check box. Click **Apply**.

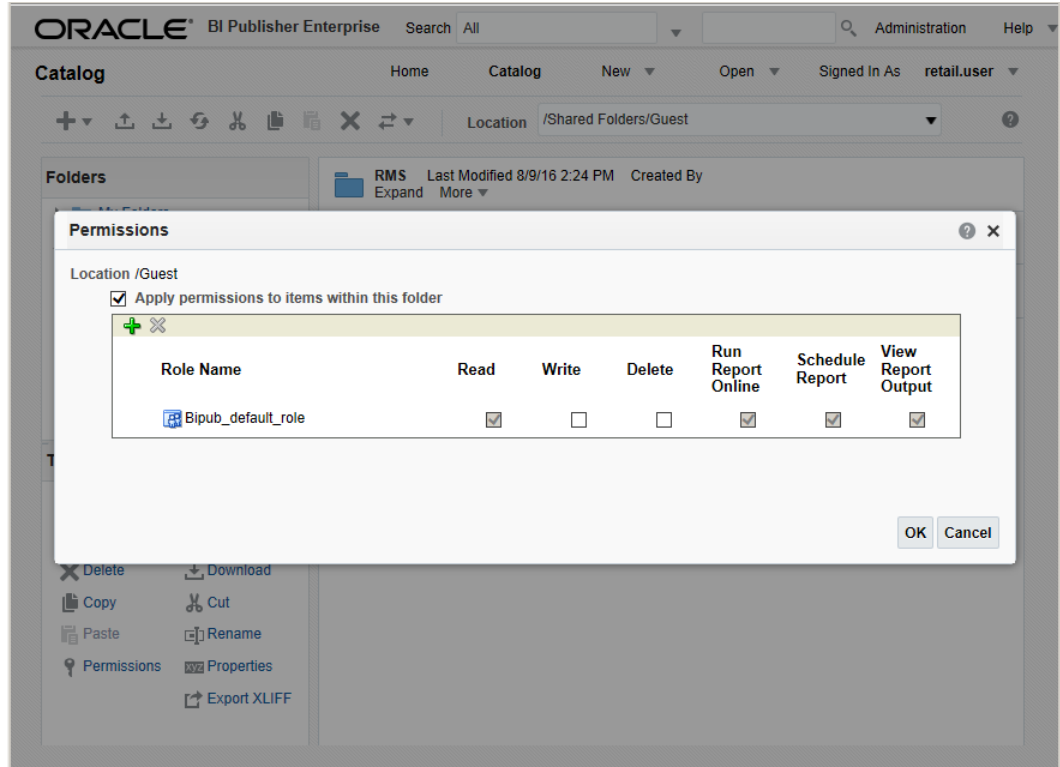
The screenshot shows the Oracle BI Publisher Administration console. The breadcrumb navigation is 'Administration > JDBC > Add Data Source'. The page title is 'Add Data Source'. There are 'Apply' and 'Cancel' buttons at the top right. The 'General' tab is selected. The page contains the following fields and options:

- Test Connection:** Button.
- Backup Data Source:** Section header.
- Use Backup Data Source:** Checked checkbox.
- Connection String:** Text input field.
- Username:** Text input field.
- Password:** Password input field.
- Test Connection:** Button.
- Security:** Section header.
- Allow Guest Access:** Checked checkbox.
- Allowed User:** Text input field.
- Available Rules:** List of rules with 'Move' and 'Remove All' buttons.
- Allowed Rules:** List of rules with 'Move All', 'Remove', and 'Remove All' buttons.

8. Click **Catalog** link at the top of the screen – and then click the **Guest** folder on the left so that it is highlighted.



9. Click the Permissions link on the lower left of the screen.



10. Click **OK**.

11. Restart WebLogic Server.

Configuring Forms to use BI Publisher

This section provides information about how to configure forms for BI.

Check the Configuration of formsweb.cfg

The formsweb.cfg file is modified by the application installer to add the appropriate entries for the RWMS Forms configurations. Please validate the file to ensure that the following items have been updated accordingly by the installer. In italics and highlighted are post installation configuration changes.

The file is typically found within the WebLogic domain configuration folder in the subdirectory
<DOMAIN_HOME>/config/fmwconfig/servers/<WLS_INSTANCE>/applications/formsapp_12.2.1/config directory and is named from the <INSTALLATION_NAME> value.

For example:

```
/u00/webadmin/config/domains/wls12c_forms/FORMSDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config
```

The following is a sample of URL definitions that were created by the installer from the formsweb.cfg file.

```
[rwmsinst]
envfile=../rwmsinst/rwmsinst.env
width=950
height=685
separateFrame=true
lookAndFeel=Oracle
colorScheme=swan
archive=frmall.jar,rwms-icons.jar
basejnlp=base.jnlp
webstart=enabled
form=logon_scr.fmx
userid=@rwmsinst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwmsinst_hh]
envfile=../rwmsinst/rwmsinst.env
width=100%
height=100%
separateFrame=false
basejnlp=base.jnlp
webstart=enabled
form=hh_intro_s.fmx
lookAndFeel=Oracle
Logo=false
colorScheme=swan
background=false
ShowMenuBar=false
>ShowStatusBar=false
baseHTMLjiniator=basejini.htm
baseHTML=base.htm
otherparams=term=/u00/webadmin/config/domains/wls12c_forms/FORMSDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config/rwmsinst/fmrweb.res
usesdi=YES
```

```

splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0" marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=@rwmsinst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwmsinst_tm]
envfile=./rwmsinst/rwmsinst.env
width=100%
height=100%
separateFrame=false
basejnlp=base.jnlp
webstart=enabled
form=tm_intro_s.fmx
Logo=false
lookAndFeel=Oracle
background=false
colorScheme=swan
ShowMenuBar=false
ShowStatusBar=false
baseHTMLjinitiator=basejini.htm
baseHTML=base.htm
otherparams=term=/u00/webadmin/config/domains/wls12c_forms/FORMSDomain/config/fmwc
onfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config/rwmsinst/fmrweb.res
usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0" marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=@rwmsinst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwmsinst_wr]
envfile=./rwmsinst/rwmsinst.env
width=100%
height=100%
separateFrame=false
basejnlp=base.jnlp
webstart=enabled
form=wr_intro_s.fmx
Logo=false
background=false
lookAndFeel=Oracle
colorScheme=swan
ShowMenuBar=false
ShowStatusBar=false
baseHTMLjinitiator=basejini.htm
baseHTML=base.htm
otherparams=term=/u00/webadmin/config/domains/wls12c_forms/FORMSDomain/config/fmwc
onfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config/rwmsinst/fmrweb.res
usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0" marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=@rwmsinst
#ssoMode=true
#ssoDynamicresourceCreate=true

```

Note: The installer inserts a splashScreen= line at three locations in the formsweb.cfg file. Having these lines in the formsweb.cfg file may sometimes cause issues and should be commented out (#splashScreen=false).

Configuring the Forms Installation Environment Files

This section outlines the entries that need to be updated in the <INSTALLATION_NAME>.env file. Please validate the file to ensure that the following items have been updated accordingly. Update variables that have not already been updated by the installer. The file is typically found within the WebLogic domain configuration folder in the subdirectory <DOMAIN_HOME>/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config directory and is named from the <INSTALLATION_NAME> value. For example:

```
/u00/webadmin/config/domains/wls12c_forms/FORMSDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_12.2.1/config
```

The following are an explanation of some variables that will be added to the <rwms_installation_name>.env (Example: rwms15inst.env) file by the rwms installer.

Field Title	RWMS_REPORTS_URL
Field Description	This field points to BI Publisher URL location for the RWMS reports repository. RWMS Installer does not update this. This is a required manual entry.
Example	http://<bip_hostname>:<Port>/xmlpserver/servlet/report?f=/Guest/RWMS/RWMS+REPOSITORY

Field Title	RWMS_REPORTS_TEMP
Field Description	This is for specifying a subdirectory of the application war files. It is used as a drop off location for reports generated in BI publisher so that the application server can pick them up and display them to the users. RWMS Installer does not update this. This is a required manual entry. The trailing sub directories for the RWMS_REPORTS_TEMP variable after .../war in the example below must exactly match with the directories trailing the /forms/that is set by the RWMS installer for RWMS_REPTEMP_ALIAS.
Example	/u00/webadmin/config/domains/wls_forms/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_12.2.1/pgdzbo/war/rwms/reptemp

Field Title	RWMS_REPORT_END_POINT_URL
Field Description	This field is used by RWMS forms to access BI Publisher's ReportService web service, which is required for emailing and executing reports. RWMS Installer does not update this. This is a required manual entry.
Example	http://<bip_hostname>:<Port>/xmlpserver/services/v2/ReportService?wsdl

Field Title	RWMS_SCHEDULE_END_POINT_URL
Field Description	This field is used by RWMS forms to access BI Publisher's ScheduleService web service, which is required for emailing and executing reports. RWMS Installer does not update this. This is a required manual entry.
Example	http://<bip_hostname>:<Port>/xmlpserver/services/v2/ScheduleService?wsdl

Field Title	RWMS_BI_USER
Field Description	The installer now uses the Java wallet files for passing the correct credentials to BI Publisher. This is the account ID of the BI User account. It is not used if the wallet is in use. (RWMS_WALLET_LOGON=TRUE) Note: The RWMS installer by default does not create this entry in the .env file. It is used as a literal string when the other .env file variable RWMS_WALLET_LOGON=FALSE and would need to be manually added.
Example	<BI Publisher user id>

Field Title	RWMS_BI_PWD
Field Description	The installer already correctly sets the alias so no changes are necessary. It is used to pass credentials to BI Publisher via its web services. Note: RWMS installer does not create the wallet. The java wallet that needs to be created in order to use this will need to have a matching alias. Note: If the .env file variable RWMS_WALLET_LOGON is set to FALSE this variable will be used by the application as a literal string and needs to be set as well as RWMS_BI_USER in order for BI Publisher to work.
Example	Wallet Example: BI_ALIAS Non wallet usage: <password>

Field Title	RWMS_DB_CONNECT
Field Description	<p>This is set to a Java wallet alias by the installer. It is used in the purging of cont_labels_to_print table by passing a fully qualified database connection when labels are printed via the RWMS application.</p> <p>Note: The Java wallet used for this variable is not setup by the installer. See RWMS Wallet Configurations in this install guide for information on how to set it up.</p> <p>Note: If the .env file variable RWMS_WALLET_LOGON is set to FALSE this variable will be used by the application as a literal string.</p>
Example	<p>Wallet Example: <schema_owner>_<db_servicename></p> <p>Non wallet usage: <dbuser>/<password>@<db_servicename></p>

Field Title	RWMS_WALLET_PATH
Field Description	<p>This variable is used by the application to determine where the java wallet file is located at so that the application can access the RWMS_BI_USER, RWMS_BI_PWD, and RWMS_DB_CONNECT values.</p> <p>The installer will create a value for this variable.</p> <p>Note: The Java wallet used for this variable is not setup by the installer. See RWMS Wallet Configurations in this install guide for information on how to set it up.</p>
Example	/projects/rwms15/extras/javawallet

Field Title	RWMS_WALLET_PARTITION
Field Description	<p>This variable is used by the application to determine what partition is to be used for the java wallet that the RWMS application uses so that the application can access the RWMS_BI_USER, RWMS_BI_PWD, and RWMS_DB_CONNECT values.</p> <p>The RWMS installer will create a value for this variable.</p> <p>Note: The Java wallet used for this variable is not setup by the installer. See RWMS Wallet Configurations in this install guide for information on how to set it up.</p>
Example	rwms15inst

Field Title	RWMS_WALLET_LOGON
Field Description	<p>This variable is used by the application to determine if it should consider the RWMS_BI_USER – RWMS_BI_PWD – and RWMS_DB_CONNECT variables as wallet aliases or string values.</p> <p>By default the installer set up variables in the .env file for java wallet usage, but the wallet file must be created as a post installation step. There are detailed instructions in the Appendix of this installation guide on how to do this.</p>
Example	TRUE or FALSE

Field Title	RWMS_SYS_REPORTS_DIR
Field Description	This variable is a reference to the Guest sub directory within BI Publisher in which the reports reside that the RWMS application accesses via the BI Publisher web service interface. RWMS Installer does not update this. This is a required manual entry.
Example	/Guest/RWMS/RWMS SYSTEM REPOSITORY/

Field Title	RWMS_FORMS_SERVER
Field Description	The RWMS installer creates this entry. The RWMS application at run time will concatenate this value with the RWMS_REPTEMP_ALIAS value and render the report to the end user in a web browser. The value is the forms server and port portion of the forms application URL.
Example	<a href="http://<formshostname>:<port>">http://<formshostname>:<port>

Field Title	RWMS_REPTEMP_ALIAS
Field Description	The RWMS installer creates this entry. The RWMS application at run time will concatenate this value with the RWMS_FORMS_SERVER value and render the report to the end user in a web browser. The value that it is set to is related to the forms war directory where it can be accessed and displayed by the forms server. The trailing sub directories for the RWMS_REPORTS_TEMP variable after .../war must exactly match with the directories trailing the /forms/ portion of the example below. This variable is used by the forms application server so that it knows where to pick the reports when the forms server renders the report to the user.
Example	/forms/rwms/reptemp/

Field Title	ORACLE_RWMS_EXTRAS
Field Description	The RWMS installer creates this entry. It is where the installer will also place the following jar files. httputil.jar, bihelper.jar, wmsSecurity.jar, and retail-public-security-api.jar.
Example	<RETAIL_HOME>/extras

Field Title	RF_LAUNCH_VALUE_ALIAS
Field Description	The RWMS installer creates this entry. This variable is used by the forms application server to find the RF launch screen modules and render them to the user. The RWMS application at run time will concatenate this value with the RWMS_FORMS_SERVER value and render the RF menu to end users on the terminal server. The Adding Radio Frequency Launch Configurations section of this install guide explains file placement for rf_launch modules and configuration.
Example	/forms/rwms/rf_launch/

Field Title	CLASSPATH
Field Description	The following jars located in RETAIL_HOME/extras are added to the CLASSPATH by the installer bihelper.jar wmsSecurity.jar retail-public-security-api.jar Verify that they were added.

The following is an example of the entries used for a working <INSTALLATION_NAME>.env file:

```
CLASSPATH=/scratch/u00/webadmin/products/wls_forms/jlib/frmbipc.jar:/scratch/u00/webadmin/products/wls_forms/forms/j2ee/frmsrv.jar:/scratch/u00/webadmin/products/wls_forms/forms/provision/frmconfig.jar:/scratch/u00/webadmin/products/wls_forms/jlib/ldapjclnt11.jar:/scratch/u00/webadmin/products/wls_forms/jlib/debugger.jar:/scratch/u00/webadmin/products/wls_forms/jlib/ewt3.jar:/scratch/u00/webadmin/products/wls_forms/oracle_common/modules/oracle.bali.share/share.jar:/scratch/u00/webadmin/products/wls_forms/jlib/utj.jar:/scratch/u00/webadmin/products/wls_forms/jlib/zrclient.jar:/scratch/u00/webadmin/products/wls_forms/reports/jlib/rwrun.jar:/scratch/u00/webadmin/products/wls_forms/forms/java/frmwebutil.jar:/scratch/u00/webadmin/products/wls_forms/oracle_common/modules/oracle.jps/jps-manifest.jar:/scratch/u00/webadmin/products/wls_forms/oracle_common/modules/oracle.dms/dms.jar:/scratch/u00/webadmin/products/wls_forms/oracle_common/modules/oracle.odl/ojdl.jar:/scratch/u00/webadmin/products/wls_forms/oracle_common/modules/javax.management.j2ee.jar:/vol.rtk/pkg_mocks/rwms160/premock/APP/retail_home/extras/bihelper.jar:/vol.rtk/pkg_mocks/rwms160/premock/APP/retail_home/extras/wmsSecurity.jar:/vol.rtk/pkg_mocks/rwms160/premock/APP/retail_home/extras/retail-public-security-api.jar
```

...

...

#RWMS Application Server added lines

```
NLS_DATE_FORMAT=DD-MON-RR
NLS_LANG=AMERICAN_AMERICA.AL32UTF8
```

```
FORMS_USERNAME_CASESENSITIVE=1
FORMS_REJECT_GO_DISABLED_ITEM=FALSE
```



```

RWMS_SYS_REPORTS_DIR=/Guest/RWMS/RWMS SYSTEM REPOSITORY/

RWMS_REPORTS_URL=http://bihostname:29704/xmlpserver/servlet/report?f=/Guest/RWMS/RWMS+REPOSITORY
RWMS_REPORTS_TEMP=/u00/webadmin/config/domains/wls12c_forms/FORMSDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_12.2.1/pgdzbo/war/rwms/reptemp

ORACLE_RWMS_EXTRAS=/u00/projects/rwms16.0/devlinf/extras
RWMS_FORMS_SERVER=http://formshostname:9001
RWMS_REPTEMP_ALIAS=/forms/rwms/reptemp/
RF_LAUNCH_VALUE_ALIAS=/forms/rwms/rf_launch/

RWMS_WALLET_PATH= /projects/rwms/extras/javawallet
RWMS_WALLET_PARTITION=rwmsinst
RWMS_WALLET_LOGON=TRUE
RWMS_BI_PWD=BI_ALIAS
RWMS_DB_CONNECT=wms01_mydb
RWMS_REPORT_END_POINT_URL=http://bihostname:29704/xmlpserver/services/v2/ReportService?wsdl
RWMS_SCHEDULE_END_POINT_URL=http://bihostname:29704/xmlpserver/services/v2/ScheduleService?wsdl
RWMS_ADF_URL=https://rwms_apphostname:7162/rwms/faces/formshandoff
...

```

If not using a wallet to configure the variables used to integrate BIPublisher and RWMS the following is an example of how the impacted entries are used for a working <INSTALLATION_NAME>.env file:

```

#RWMS_WALLET_PATH=/projects/rwms/extras/javawallet
#RWMS_WALLET_PARTITION=rwmsinst
RWMS_WALLET_LOGON=FALSE
RWMS_BI_USER=retail.user
RWMS_BI_PWD=welcme1
RWMS_DB_CONNECT=wms01/retail@mydb

```

Update <INSTALLATION_NAME>.env File With FORMS_DATETIME_LOCAL_TZ

Within the RWMS application the Working Days Editor screen displays the following message "The entry does not match the valid set or range." in some cases where end times is actually within a valid range. In addition to that, it saves inaccurate times in WORKING_DAYS table. In order to resolve these issues the FORMS_DATETIME_LOCAL_TZ variable must be added to the <INSTALLATION_NAME>.env file.

Example:

```
FORMS_DATETIME_LOCAL_TZ=GMT
```

RWMS Wallet Configurations

The RWMS application is comprised of several entities that require logon credentials to be passed back and forth. These credentials are kept secure by the use of several different types of wallets. The reason that multiple wallets are required is that some of the RWMS entities are from different tech stacks. The RWMS application needs to allow ADF, Oracle Forms, BIPublisher, and batch script interfaces to be able to communicate. Below is a summary of each type of wallet that is utilized and how it works in the application.

Application (Java) Wallet

Note: This wallet must be created manually. The installer does not create it.

- Its purpose is to provide credentials for the RWMS application to be able to print reports via web service connections to in BI Publisher.
- This wallet is not setup by RWMS installer.
- See [Setting up Application \(Java\) Wallet for RWMS .env File](#) for instructions on how to set this up.
- Used to store values for the variables setup in the RWMS forms .env file.
 - RWMS_WALLET_PATH:
 - By default the RWMS installer creates this entry in the .env file, but does not actually create the wallet.
 - The path here should be set to the location where the wallet will reside once it is created. The location is a suggested location, but can actually reside anywhere on the file system that the RWMS application has access to.
 - RWMS_WALLET_PARTITION:
 - By default the RWMS installer creates this entry in the .env file, but does not actually create the wallet entry.
 - This value is the partition of the wallet that is used to look up the credentials.
 - RWMS_WALLET_LOGON:
 - By default the RWMS installer creates this entry in the .env file as TRUE.
 - This value tells the RWMS application whether or not to consider the variables
 - Setting this to FALSE, commenting it out, or deleting the entry will be accepted as a FALSE value by the application and will result in RWMS application considering credential variables as clear text.
 - RWMS_BI_USER:
 - No alias for this variable is stored in the wallet. The application uses the RWMS_BI_PWD for this purpose.
 - Used to store a BI Publisher username credential for RWMS application to access BI Publisher reports via web services.
 - Only used in when RWMS_WALLET_LOGON is set to FALSE and will be in clear text.
 - RWMS_BI_PWD:
 - By default the RWMS installer creates this entry in the .env file, but does not actually create the wallet entry.
 - The variable value is an alias that access the credentials used to log into BI Publisher in order to print reports from the application.
 - When RWMS_WALLET_LOGON is set to FALSE – this value should be clear text in the .env file.
 - RWMS_DB_CONNECT:
 - By default the RWMS installer creates this entry in the .env file, but does not actually create the wallet entry.
 - The variable value is an alias that access the credentials used to create a connection to the database by the purge_cont_labels_to_print_b.sh batch

script. It is used to clean up the cont_labels_to_print table after labels are printed from the RWMS application.

- When RWMS_WALLET_LOGON is set to FALSE – this value should be clear text in the .env file.

Oracle Wallet

- Its purpose is to provide a secure connection string for RWMS in the formsweb.cfg file for automatic forms launching, and executing batch programs via command line or daemon processes.
- This wallet is setup by RWMS installer.
- See [Appendix: Setting Up an Oracle Wallet](#) for instructions on how to set this up.

WLS Database Credential Store

- Its purpose is to provide secure credentials to the file system so that the RWMS application can access and write log files.
- This wallet is setup by RWMS installer.
- See [Setting Up WLS Database Credential Store](#) for instructions on how to set this up or maintain it outside the RWMS installer.
- It is used by the Manual Script Launch Editor which uses the application.properties to acquire credentials for a file system user that can write log files on the server.
- The specific in the application.properties are as follows and are case sensitive:
 - Cred.store.map=Rwmsmap
 - Cred.store.key=rwmskey

Configuring Terminal Server for RWMS Application

The RWMS screens designed for Radio Frequency (RF) devices are accessed via a connection from the RF device to a terminal server where the application is launched upon connection. In order to provide users streamlined access and departure from the application and Terminal Server – Configurations for launching the application in kiosk mode and then terminating the users Terminal Server session will need to be completed.

Setup and Configure JNLP Launch Shortcut for Kiosk Mode

RWMS is designed to launch from a client based JNLP session due to browsers planning to drop support for Java applets in future releases. Complete the following instructions in order to set this up on the Terminal Server so that the RWMS application will launch in kiosk mode and not download a new JNLP each time.

- Log onto the Terminal Server as an administrative user
- Launch the RWMS application using each of the RF URLs that were created by the installer.

http://<formshostname>:<port>/forms/frmservlet?config=rwmsinst_hh

http://<formshostname>:<port>/forms/frmservlet?config=rwmsinst_tm

http://<formshostname>:<port>/forms/frmservlet?config=rwmsinst_wr

- For each of the URLs that is launched a .jnlp file called frmservlet.jnlp will be downloaded.
- Save the files onto the Terminal Servers file system and logically rename the files so that they reflect the corresponding RF URL.

Example:

File location: C:\develop\launch_page

Example:

Rename the frmservlet.jnlp file that was downloaded for the http://<formshostname>:<port>/forms/frmservlet?config=rwmsinst_hh as hh.jnlp and http://<formshostname>:<port>/forms/frmservlet?config=rwmsinst_tm as tm.jnlp and http://<formshostname>:<port>/forms/frmservlet?config=rwmsinst_wr as wr.jnlp

- Open each of the .jnlp files that were downloaded with a text editor and add the following parameter: `<param name="java_decorated_frame" value = "false" />`. This will remove the jnlp frame that surrounds the application and opened the application in Kiosk mode

Example:

```
...
<param name="mediaBase" value="codebase"/>
  <param name="webstartRequest" value="true" />
  <param name="java_decorated_frame" value = "false" />
  </applet-desc>
</jnlp>
...
```

- Create a shortcut for each of the .jnlp files that were created on the Terminal Server.
- Create users on the Terminal Server that would be designated for device access.

Example:

Truck Mount: Create a user called tm_user

- Copy the shortcut for the .jnlp files to the corresponding users Startup folder. This will cause the application to launch when the user logon to the Terminal Server.

Example:

C:\Users\tm_user\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\tm.jnlp - Shortcut

Configure Java Security Exceptions and Certificate Imports

Due to Java security level settings and Java versions users may be prompted with Security Dialogs to accept security risks and allow the application to run. This is not ideal due to limitations for many RF devices navigation and being able to view the prompts properly. In order to allow the application to run with prompting the user each time the application is launched – review JAVA Documentation regarding this subject by accessing the following link:

<https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/plugin003.html>

Configuring Terminal Server Close for Radio Frequency Devices

The Radio Frequency(RF) screens are accessed by the RF devices via a connection from the RF device to a terminal server where the application is launched upon connection. When users exit the terminal server the session that was opened on the Terminal server

for the RF device must be closed. This section covers the required configuration for that to occur.

File Placement

To place files, complete the following steps.

1. On the application server in the forms application server base directory you will need to create /rwms/rf_launch directories in the Forms war file directory.

Example:

```
/u00/webadmin/config/domains/wls_forms/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_12.2.1/pgdzbo/war/rwms/rf_launch
```

2. From the pickup directory listed below: Acquire and place the following files in the directory that was created during previous step.
 - Pickup directory:
`<STAGING_DIR>/rwms/installer/rwms/RF_Launch/Source/`
 - close.html
 - WindowsHelper.jar

Update the RF Launch Modules

rwms_rf_menu.html Module

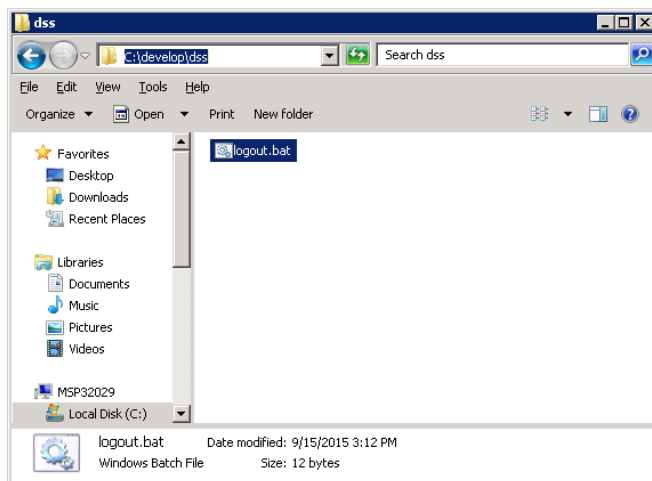
The rwms_rf_menu.html should have been updated by the installer but may be configured to launch additional URL's based on the names of the URL's in the formsweb.cfg file. The variables that are updated by the installer are listed below.

- var hh_device =
`"http://<Server_Name>:<port>/forms/frmservlet?config=<INSTALLATION_NAME>_hh";Q`
- var tm_device =
`"http://<Server_Name>:<port>/forms/frmservlet?config=<INSTALLATION_NAME>_tm";`
- var wr_device =
`"http:<Server_Name>:<port>/forms/frmservlet?config=<INSTALLATION_NAME>_wr";`
- var exit_script =
`"http://<Server_Name>:<port>/forms/rwms/rf_launch/close.html";`

close.html Module

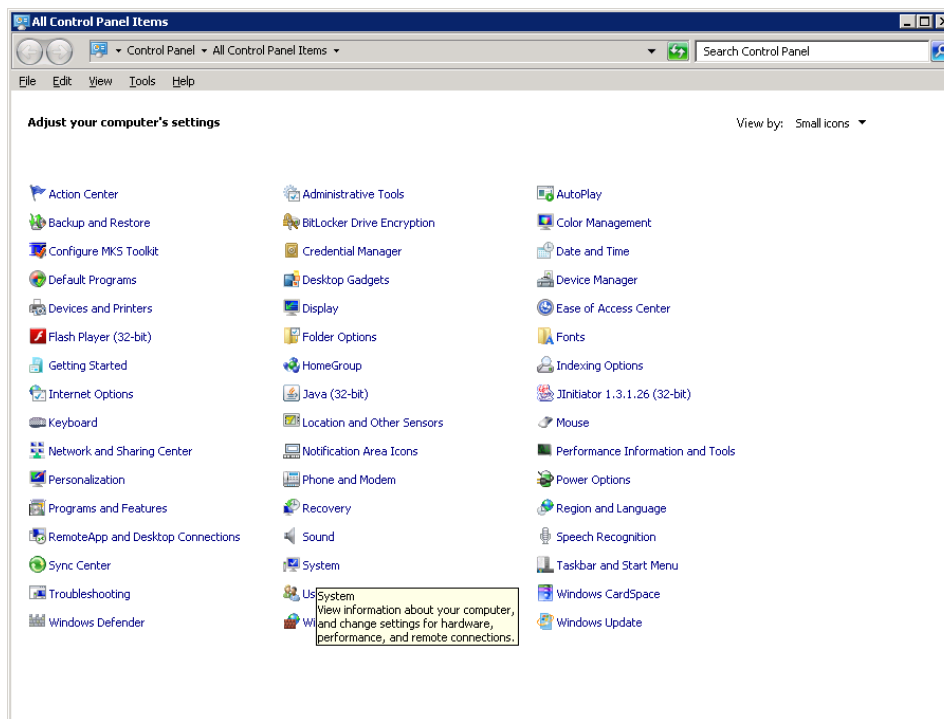
The Close.html file is designed to close down and log a user out of a Terminal Server session by running the Windows shutdown.exe. In order for this to work, manual configuration is necessary.

1. Create an executable script on the Terminal Servers file system
2. In a directory create a executable file called logout.bat

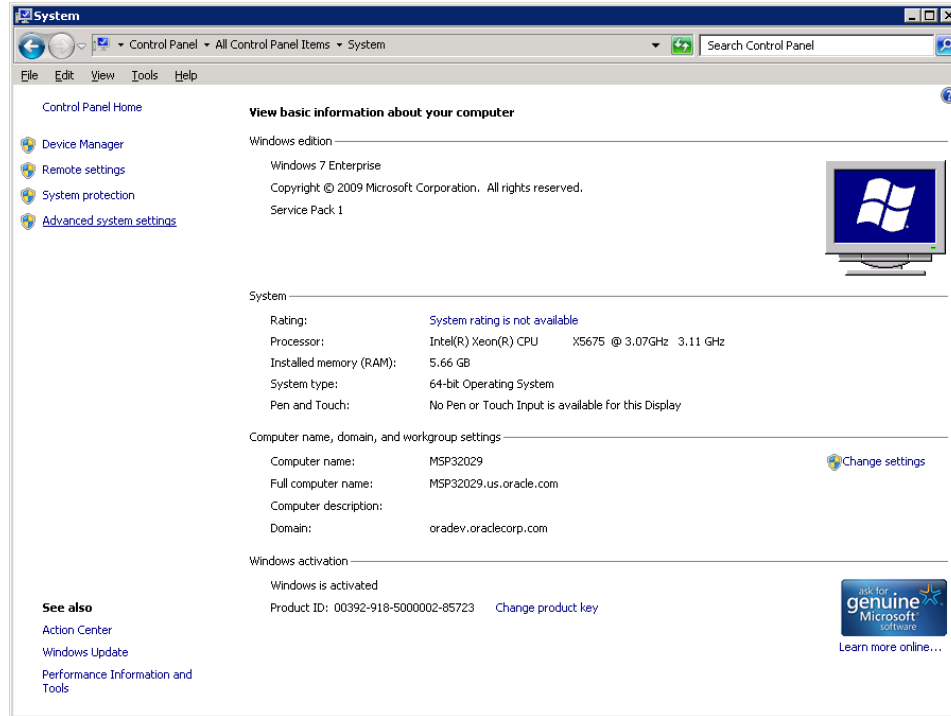


3. Right Click on the file to open it for edit and add the following:

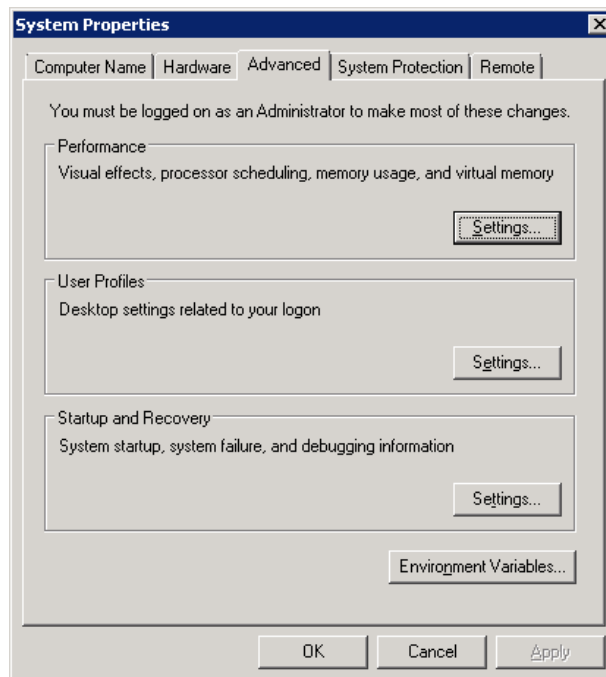
```
@shutdown /1
```
4. Save and close the file.
5. Using the Control Panel access System settings



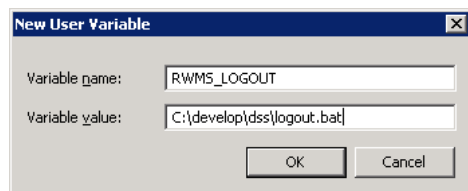
6. Click on Advanced system settings



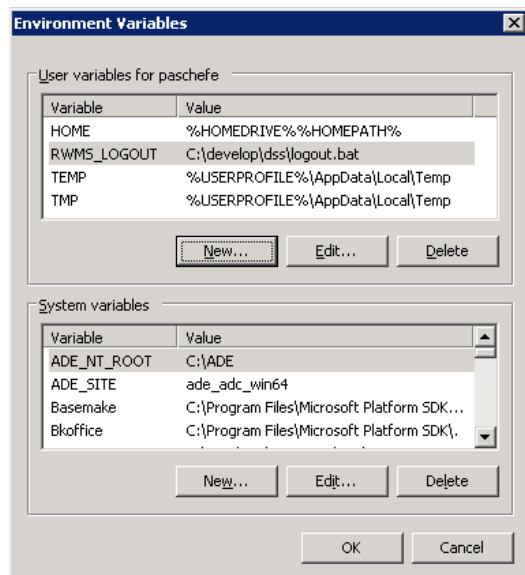
7. Under the Advanced tab – Click Environment Variables



- Click on New and create a variable called RWMS_LOGOUT. Specify the path in which you created the logout.bat file.



9. Click OK



i.

Note: Running this script on your desktop will shut it down so make sure to test it on the terminal server.

Update fmrweb.res for Keymapping

The fmrweb.res file is used to specify key-mapping for the radio frequency devices that are set up in the formsweb.cfg file.

1. Depending upon device this file may need to be updated.
2. The installer places a copy in the directory specified in the formsweb.cfg file for each radio frequency URL that is created
3. The fmrweb.res file comes with key-mapping of CTRL+<number> to work for function keys by default.

Web Services Installation

Some Oracle Retail applications; <app> (for example, RWMS) use Oracle Objects for the PL/SQL API's. The tool generates a Web Service Provider layer between the external clients and the <app> API's to provide the Web Service functionality, such as faults, logging, and security, as well as the conversion from xml payloads to Oracle Objects. The Retail Service Enabler (RSE) tool creates the appropriate Provider web service endpoints as well as templates for the PL/SQL APIs.

Note: Depending on your business needs, you may not need to install web services.

Create a Managed Server

Create a managed server for the RWMS Web services app to be deployed per the *WebLogic Installation Guide*.

Create a Datasource

Create a datasource for RWMS Webservices to point to the RWMS schema as follows.

- Name can be anything you want.
- JNDI Name must be jdbc/RetailWebServiceDs.
- Set database type and driver for your environment (use non-XA jdbc driver).
- Set connection properties for the database using the rwms user (RWMS15DEV). Be sure to test the configuration before moving on.
- Point the data source to the server created in the Create a Managed Server section above.

Deploy RWMS Service EAR File

To deploy the RWMS Service .ear file, do the following.

1. Make sure that the managed server created in Step 2, where this application will be deployed, is up and running.
2. In the left Domain Structure window, click Environment > Deployments.
3. Click **Lock and Edit** in the change center to install the ear file. It will enable the install button on the deployments screen.
4. Click **Install**.
5. Click the upload your file(s) link.
6. Click the Deployment Archive browse button.
7. Go to the location of the rwms-service.ear file. This would be <STAGING_DIR>/rwms/installer/rwms15/websevice_objects/provider/ear
8. Select the rwms-service.ear file.
9. Click **Next**. Make sure that the radio button for rwms-services.ear is selected.
10. Click **Next** again. Make sure that Install this deployment as an application is selected.
11. Click **Next** again and select the server created in Step 2.

12. Click **Next**. Click **Finish** to return to the deployments page. You should see rwms-service in the list of deployments.
13. Click **Activate Changes** in the change center. The state of the application may be shown as prepared. If so, select the check box next to rwms-service to enable the Start button. Click **Start**. Select servicing all requests.
14. To test the deployment, click on the application. Click the testing tab.

Home > Summary of Deployments > rwms-service

Settings for rwms-service

Overview | Deployment Plan | Configuration | Security | Targets | Control | **Testing** | Monitoring | Notes

Some deployment types support test points you can use to verify that a deployment was successful and that the module is ready for use. The following table includes all of the test points available for this application or module.

Deployment Tests

Name	Test Point	Comments
<input type="checkbox"/> rwms-service		
/wls-cat/index.jsp	Classloader Analysis Tool	Classloader Analysis Tool on server rwms-service-server
<input type="checkbox"/> PendingReturnsBean		There are no test points for this EJB module.
<input type="checkbox"/> PendingReturnsService		Test points for this WebService module.
<input type="checkbox"/> RwmInfrastructureManagerBean		There are no test points for this EJB module.
<input type="checkbox"/> RwmInfrastructureManagerService		Test points for this WebService module.
/RwmInfrastructureManagerBean/RwmInfrastructureManagerService	WSDL	WSDL page on server rwms-service-server
/RwmInfrastructureManagerBean/RwmInfrastructureManagerService	Test client	Test client on server rwms-service-server
<input type="checkbox"/> ShippingManifestSelectorBean		There are no test points for this EJB module.
<input type="checkbox"/> ShippingManifestSelectorService		Test points for this WebService module.

Showing 1 to 1 of 1 Previous | Next

Note: If the webservice ear is being deployed in a managed server which is running on an HTTPS port, the Test Point WSDL links may not be visible. In order to view the WSDL links, you need to temporarily enable the HTTP (non-secured) ports of Administration and the webservice managed server and login and navigate to deployments of webservice to view the WSDL links.

Configure Web Service Security

Configuring the web service deployment to use the WS-Security Username Profile involves forcing all incoming requests to contain WS Security headers to authenticate the requestor based on a user name and password elements. The use of this profile does not provide any confidentiality protection on web service requests: data contained within the Web service messages will not be encrypted. However, using a secure message transport, such as SSL/TLS, will provide confidentiality for the message as it traverses the network. For more information on using SSL/TLS see the section, "Configuring SSL" found in the WebLogic document, "Securing the WebLogic Server, 10g Release 3 (10.3)".

Additional WS Security policies may also be available depending on the configuration of the WebLogic server. Using these policies will require appropriate changes to web service requests created by applications consuming the web service. Many of these policies also require additional steps for correct keystore and truststore file configuration.

When a web service uses the WS-Security Username profile, all web service consumers must specify a user name configured within the current WebLogic domain. This user name must also have the appropriate role(s) associated with it. Using this profile is thus a two-step process:

1. Attach the WS-Security Username policy to the web service
2. Create roles and users who can access the web services

Refer to *Oracle Retail Warehouse Management System Security Guide*, the *Oracle Retail Integration Bus Security Guide*, and the *Oracle Retail Service Backbone Security Guide* for more details.

Patching Procedures

Oracle Retail Patching Process

The patching process for many Oracle Retail products has been substantially revised from prior releases. Automated tools are available to reduce the amount of manual steps when applying patches. To support and complement this automation, more information about the environment is now tracked and retained between patches. This information is used to allow subsequent patches to identify and skip changes which have already been made to the environment. For example, the patching process uses a database manifest table to skip database change scripts which have already been executed.

The enhanced product patching process incorporates the following:

- Utilities to automate the application of Oracle Retail patches to environments.
- Unified patches so that a single patch can be applied against Database, Forms, Java applications, Batch, etc. installations.
- Database and Environment manifests track versions of files at a module level.
- Centralized configuration distinguishes installation types (Database, Forms, Java, Batch, etc.).
- Patch inventory tracks the patches applied to an environment.

These enhancements make installing and updating Oracle Retail product installations easier and reduce opportunities for mistakes. Some of these changes add additional considerations to patching and maintaining Oracle Retail product environments. Additional details on these considerations are found in later sections.

Supported Products and Technologies

With version 15.0.1, several products and technologies are supported by the enhanced patching process. The utilities, processes and procedures described here are supported with the following products and listed technologies:

Product	Supported Technology
Oracle Retail Merchandising System (RMS)	<ul style="list-style-type: none"> ▪ Database scripts ▪ Batch scripts ▪ RETL scripts ▪ Data Conversion Scripts ▪ Forms ▪ BI Publisher Reports
Oracle Retail Warehouse Management System (RWMS)	<ul style="list-style-type: none"> ▪ Database scripts ▪ Batch scripts ▪ Forms ▪ BI Publisher Reports

Product	Supported Technology
Oracle Retail Price Management (RPM)	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application ▪ Batch scripts
Oracle Retail Invoice Matching (ReIM)	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application ▪ Batch scripts
Oracle Retail Allocation	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application ▪ Batch scripts
Oracle Retail Sales Audit (ReSA)	<ul style="list-style-type: none"> ▪ Database scripts (included with RMS) ▪ Java Application
Oracle Retail Insights (RI) Previously called Oracle Retail Analytics (RA)	<ul style="list-style-type: none"> ▪ Database scripts
Oracle Retail Advanced Science Engine (ORASE)	<ul style="list-style-type: none"> ▪ Database scripts ▪ Batch scripts
Oracle Retail Data Extractor (RDE)	<ul style="list-style-type: none"> ▪ Database scripts
Oracle Retail Application Admin Console (ORAAC). Previously called Oracle Retail Application Security Role Manager (RASRM)	<ul style="list-style-type: none"> ▪ Java Application

Patch Concepts

During the lifecycle of an Oracle Retail environment, patches are applied to maintain your system. This maintenance may be necessary to resolve a specific issue, add new functionality, update to the latest patch level, add support for new technologies, or other reasons.

A patch refers to a collection of files to apply to an environment. Patches could be cumulative, such as the 15.0.1 release, or incremental, such as a hot fix for just a few modules. Patches may contain updates for some or all components of a product installation including database, application code, forms, and batch. In a distributed architecture the same patch may need to be applied to multiple systems in order to patch all of the components. For example, if a patch contains both database and application changes, the patch would need to be applied to both the database server and the application server.

The top-level directory for the installation of an Oracle Retail product is referred to as the `RETAIL_HOME`. Underneath `RETAIL_HOME` are all of the files related to that product installation, as well as configuration and metadata necessary for the Oracle Retail Patch Assistant to maintain those files. In some cases the runtime application files also exist under `RETAIL_HOME`. For example, the compiled RMS forms, compiled RMS batch files, or Java Application batch scripts.

Patching Utility Overview

Patches are applied and tracked using utilities that are specifically designed for this purpose. The primary utility is described briefly below and additional information is available in later sections.

Oracle Retail Patch Assistant (ORPatch)

ORPatch is the utility used to apply patches to an Oracle Retail product installation. It is used in the background by the installer when creating a new installation or applying a cumulative patch. It is used directly to apply an incremental patch to an environment.

Oracle Retail Merge Patch (ORMerge)

ORMerge is a utility to allow multiple patches to be combined into a single patch. Applying patches individually may require some steps to be repeated. Merging multiple patches together allows these steps to be run only once. For example, applying several incremental patches to database packages will recompile invalid objects with each patch. Merging the patches into a single patch before applying them will allow invalid objects to be recompiled only once.

Oracle Retail Compile Patch (ORCompile)

ORCompile is a utility to compile components of Oracle Retail products outside of a patch. It allows RMS Forms, RMS Batch, and RWMS Forms to be fully recompiled even if no patch has been applied. It also contains functionality to recompile invalid database objects in product schemas.

Oracle Retail Deploy Patch (ORDeploy)

ORDeploy is a utility to deploy components of Oracle Retail Java products outside of a patch. It allows RPM, ReIM, Allocation and ReSA java applications to be redeployed to WebLogic even if a patch has not been applied. It contains functionality to optionally include or not include Java customizations when redeploying.

Changes with 15.0

Some products and technologies are supported by the enhanced patching process for the first time in 15.0. In those cases all of the content in this chapter is new with 15.0.

Renamed Products

For the 15.0 release several products using Orpatch have been renamed. Oracle Retail Analytics (RA) is renamed to Oracle Retail Insights (RI) and Oracle Retail Application Security Role Manager (RASRM) is renamed to Oracle Retail Application Admin Console (ORAAC). The old product names are still used in the underlying Orpatch code, the Orpatch Action names, the Orpatch env_info.cfg, and the Orpatch wallet paths. Documentation for these products will refer to the new names.

Patching Considerations

Patch Types

Oracle Retail produces two types of patches for their products: cumulative and incremental.

Cumulative Patches

A cumulative patch includes all of the files necessary to patch an environment to a specific level or build a new environment at that level. Examples of cumulative patches would be 15.0.1, 15.0.2, and so on. Cumulative patches come with a standard Oracle Retail installer and so can be applied to an environment with the installer rather than with ORPatch or other utilities.

Incremental Patches

An incremental patch includes only selected files necessary to address a specific issue or add a feature. Examples of incremental patches would be a hot fix for a specific defect. Incremental patches do not include an installer and must be applied with ORPatch.

Incremental Patch Structure

An Oracle Retail incremental patch generally contains several files and one or more subdirectories. The subdirectories contain the contents of the patch, while the individual files contain information about the patch and metadata necessary for patching utilities to correctly apply the patch. The most important files in the top-level directory are the README.txt, the manifest files.

README File

The README.txt file contains information about the incremental patch and how to apply it. This may include manual steps that are necessary before, after or while applying the patch. It will also contain instructions on applying the patch with ORPatch.

Manifest Files

Each patch contains manifest files which contain metadata about the contents of a patch and are used by ORPatch to determine the actions necessary to apply a patch. Patches should generally be run against all installations a product in an environment, and ORPatch will only apply the changes from the patch that are relevant to that installation.

Note: Cumulative patches use a different patch structure because they include a full installer which will run ORPatch automatically.

Version Tracking

The patching infrastructure for 15.0 tracks version information for all files involved with a product installation. The RETAIL_HOME contains files which track the revision of all files within the RETAIL_HOME including batch, forms, database, Java archives and other files. In addition, records of database scripts that have been applied to the product database objects are kept within each database schema.

Apply all Patches with Installer or ORPatch

In order to ensure that environment metadata is accurate all patches must be applied to the Oracle Retail product installation using patching utilities. For cumulative patches this is done automatically by the installer. For incremental patches ORPatch must be used directly. This is especially important if database changes are being applied, in order to ensure that the database-related metadata is kept up-to-date.

Environment Configuration

A configuration file in \$RETAIL_HOME/orpatch/config/env_info.cfg is used to define the details of a specific Oracle Retail environment. This file defines:

- The location of critical infrastructure components such as the ORACLE_HOME on a database or middleware server.
- The location of Oracle Wallets to support connecting to the database users.
- The type of file processing which is relevant to a particular host. For example, if this is a host where database work should be done, or a host where batch compilation should be done, a host where Java applications should be deployed, etc. This allows a single database, forms and batch patch to be run against all types of hosts, applying only the relevant pieces on each server.
- Other configuration necessary to determine proper behavior in an environment.

Retained Installation Files

The RETAIL_HOME location of an Oracle Retail product installation contains all of the files associated with that installation. This can include database scripts, Java files, Forms, Batch, RETL and Data Conversion files as with previous versions and also includes all database scripts. This allows objects to be reloaded during patching, including any necessary dependencies.

Reloading Content

In order to ensure that database contents and generated files exactly match patched versions, when applying cumulative patches some content is regenerated even if it does not appear to have changed.

On a cumulative patch this includes:

- All re-runnable database content will be reloaded
 - Packages and Procedures
 - Database Types (excluding RIB objects)
 - Control scripts
 - Triggers
 - Webservice jars and packages
 - Form Elements
- All RMS and RWMS forms files will be recompiled
- All RMS batch files will be recompiled

When applying incremental patches, only changed files will be reloaded. However this does not apply to RMS batch, which is fully recompiled with any change.

Java Hotfixes and Cumulative Patches

When applying cumulative patches to Java applications components with ORPatch, all hotfixes related to base product ear files included with the patch will be rolled back. This increases the likelihood of a successful deployment because hotfixes may not be compatible with updated product ear files, or may already be included with the ear. Before applying a cumulative patch to Java applications, check the patch documentation to determine which hotfixes are not included in the ear. Then work with Oracle Support to obtain compatible versions of the fixes for the updated ear version. In some cases this may be the same hotfix, in which case it can be re-applied to the environment. In other cases a new hotfix may be required.

Backups

Before applying a patch to an environment, it is extremely important to take a full backup of both the RETAIL_HOME file system and the Oracle Retail database. Although ORPatch makes backups of files modified during patching, any database changes cannot be reversed. If a patch fails which contains database changes, and cannot be completed, the environment must be restored from backup.

Disk Space

When patches are applied to an environment, the old version of files which are updated or deleted are backed up to \$RETAIL_HOME/backups/backup-<timestamp>. When applying large patches, ensure there is sufficient disk space on the system where you unzip the patch or the patching process may fail. Up to twice as much disk space as the unzipped patch may be required during patching.

In addition to backups of source files, the existing compiled RMS or RWMS Forms and RMS Batch files are saved before recompilation. These backups may be created during patches:

- Batch 'lib' directory in \$RETAIL_HOME/oracle/lib/bin-<timestamp>
- Batch 'proc' directory in \$RETAIL_HOME/oracle/proc/bin-<timestamp>
- Forms 'toolset' directory in \$RETAIL_HOME/base/toolset/bin-<timestamp>
- Forms 'forms' directory in \$RETAIL_HOME/base/forms/bin-<timestamp>

Periodically both types of backup files can be removed to preserve disk space.

Patching Operations

Running ORPatch

ORPatch is used to apply patches to an Oracle Retail product installation. When applying a patch which includes an installer, ORPatch does not need to be executed manually as the installer will run it automatically as part of the installation process. When applying a patch that does not include an installer, ORPatch is run directly.

ORPatch performs the tasks necessary to apply the patch:

- Inspects the patch metadata to determine the patch contents and patch type.
- Reads the environment configuration file to determine which product components exist in this installation.
- Assembles a list of patch actions which will be run on this host to process the patch.
- Executes pre-checks to validate that all patch actions have the necessary configuration to proceed.
- Compares version numbers of files from the patch against the files in the environment.
- Backs up files which will be updated.
- Copies updated files into the installation.
- Loads updated files into database schemas, if applicable.
- Recompiles RMS batch, if applicable.
- Recompiles RMS forms, if applicable.
- Recompiles RWMS forms, if applicable.
- Constructs updated Java archives and deploys them to WebLogic, if applicable
- Updates Java batch files and libraries, if applicable
- Records the patch in the patch inventory.

If a patch does not contain updated files for the database or system, no action may be taken. If a previously failed ORPatch session is discovered, it will be restarted.

Preparing for Patching

Before applying a patch to your system, it is important to properly prepare the environment.

Single Patching Session

It is extremely important that only a single ORPatch session is active against a product installation at a time. If multiple patches need to be applied, you can optionally merge them into a single patch and apply one patch to the environment. Never apply multiple patches at the same time.

Shutdown Applications

If a patch updates database objects, it is important that all applications are shutdown to ensure no database objects are locked or in use. This is especially important when applying changes to Oracle Retail Integration Bus (RIB) objects as types in use will not be correctly replaced, leading to “ORA-21700: object does not exist or marked for delete” errors when restarting the RIB.

Backup Environment

Before applying a patch to an environment, it is important to take a full backup of both the RETAIL_HOME file system and the retail database. Although ORPatch makes backups of files modified during patching, any database changes cannot be reversed. If a patch which contains database changes fails and cannot be completed, the environment must be restored from backup.

Log Files

When applying a patch, ORPatch will create a number of log files which contain important information about the actions taken during a patch and may contain more information in the event of problems. Log files are created in the \$RETAIL_HOME/orpatch/logs directory. Logs should always be reviewed after a patch is applied.

After a patch session the log directory will contain at a minimum an ORPatch log file and may also contain other logs depending on the actions taken. The following table describes logs that may exist.

Log File	Used For
orpatch-<date>-<time>.log	Primary ORPatch log file
detail_logs/dbsql_<component>/invalids/*	Details on the errors causing a database object to be invalid
detail_logs/analyze/details	Detail logs of files that will be created/updated/removed when a patch is applied
detail_logs/compare/details	Detail logs of the differences between two sets of environment metadata
orpatch_forms_<pid>_child_<num>.log	Temporary logs from a child process spawned to compile forms in parallel. After the child process completes, the contents are append to the primary orpatch log file
detail_logs/forms/rms_frm_toolset/*	Detail logs of the compilation of each RMS Toolset file
detail_logs/forms/rms_frm_forms/*	Detail logs of the compilation of each RMS Forms file
detail_logs/rmsbatch/lib/*	Detail logs of the compilation of RMS Batch libraries
detail_logs/rmsbatch/proc/*	Detail logs of the compilation of RMS Batch programs
detail_logs/dbsql_rms/rms_db_ws_consumer_jars/*	Detail logs of the loadjava command to install RMS WebService Consumer objects
detail_logs/dbsql_rms/rms_db_ws_consumer_libs/*	Detail logs of the loadjava command to install RMS WebService Consumer libraries

Log File	Used For
detail_logs/forms/rwms_frm_forms/*	Detail logs of the compilation of each RWMS Forms file
detail_logs/dbsql_rwms/rwms_db_sp_jars/*	Detail logs of the loadjava command to install RWMS SP jars
detail_logs/javaapp_<product>/deploy/*	Detail logs of the deploy of a Java product

Unzip Patch Files

Before executing ORPatch, the patch files must be unzipped into a directory. This directory will be passed to ORPatch as the “-s <source directory>” argument on the command-line when applying or analyzing a patch.

Location of ORPatch

The ORPatch script will be located in \$RETAIL_HOME/orpatch/bin.

Command Line Arguments

ORPatch behavior is controlled by several command-line arguments. These arguments may be actions or options. Command and option names can be specified in upper or lower case, and will be converted to upper-case automatically. Arguments to options, for example the source directory patch, will not be modified.

ORPatch command-line actions:

Action	Description
apply	Tells ORPatch to apply a patch, requires the -s option Example: orpatch apply -s \$RETAIL_HOME/stage/patch123456
analyze	Tells ORPatch to analyze a patch, requires the -s option Example: orpatch analyze -s \$RETAIL_HOME/stage/patch123456
lsinventory	Tells ORPatch to list the inventory of patches that have been applied to this installation
exportmetadata	Tells ORPatch to extract all metadata information from the environment and create a \$RETAIL_HOME/support directory to contain it. Requires the -expname option.
diffmetadata	Tells ORPatch to compare all metadata from the current environment with metadata exported from some other environment. Requires the -expname and -srname options.
revert	Tells ORPatch to revert the files related to a patch, requires the -s option Example: orpatch revert -s \$RETAIL_HOME/backups/backup-09302013-153010

Note: An action is required and only one action can be specified at a time.

ORPatch command-line arguments:

Argument	Valid For Actions	Description
-s <source dir>	apply analyze	Specifies where to find the top-level directory of the patch to apply or analyze. The source directory should contain the manifest.csv and patch_info.cfg files.
-new	apply	Forces ORPatch to not attempt to restart a failed ORPatch session
-expname	exportmetadata diffmetadata lsinventory	Defines the top-level name to be used for the export or comparison of environment metadata. When used with lsinventory, it allows an exported inventory to be printed.
-srcname	diffmetadata	Defines the 'name' to use when referring to the current environment during metadata comparisons.
-dbmodules	diffmetadata	When comparing metadata at a module-level, compare the dbmanifest information rather than the environment manifest. This method of comparing metadata is less accurate as it does not include non-database files.
-jarmodules	analyze diffmetadata	When used with analyze, requests a full comparison of the metadata of Java archives included in the patch versus the metadata of the Java archives in the environment. This behavior is automatically enabled when Java customizations are detected in the environment. Analyzing the contents of Java archives allows for detailed investigation of the potential impacts of installing a new Java ear to an environment with customizations. When used with diffmetadata, causes metadata to be compared using jarmanifest information rather than the environment manifest. This provides more detailed information on the exact differences of the content of Java archives, but does not include non-Java files.
-selfonly	apply analyze	Only apply or analyze changes in a patch that relate to orpatch itself. This is useful for applying updates to orpatch without applying the entire patch to an environment.
-s <backup dir>	revert	Specifies the backup from a patch that should be reverted to the environment. This restores only the files modified during the patch, the database must be restored separately or the environment will be out-of-sync and likely unusable.

Analyzing the Impact of a Patch

In some cases, it may be desirable to see a list of the files that will be updated by a patch, particularly if files in the environment have been customized. ORPatch has an 'analyze' mode that will evaluate all files in the patch against the environment and report on the files that will be updated based on the patch.

To run ORPatch in analyze mode, include 'analyze' on the command line. It performs the following actions:

- Identifies files in the environment which the patch would remove.
- Compares version numbers of files in the patch to version numbers of files in the environment.
- Prints a summary of the number of files which would be created, updated or removed.
- Prints an additional list of any files that would be updated which are registered as being customized.
- Prints an additional list of any files which are in the environment and newer than the files included in the patch. These files are considered possible conflicts as the modules in the patch may not be compatible with the newer versions already installed. If you choose to apply the patch the newer versions of modules in the environment will NOT be overwritten.
- If a Java custom file tree is detected, prints a detailed analysis of the modules within Java ear files that differ from the current ear file on the system.
- Saves details of the files that will be impacted in `$RETAIL_HOME/orpatch/logs/detail_logs/analyze/details`.

This list of files can then be used to assess the impact of a patch on your environment.

To analyze a patch, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Set the JAVA_HOME environment variable if the patch contains Java application files.

```
export JAVA_HOME=/u00/oretail/java_jdk
```

Note: If the JAVA_HOME environment variable is not specified, the value from `RETAIL_HOME/orpatch/config/env_info.cfg` will be used.

5. Create a staging directory to contain the patch, if it does not already exist.

```
mkdir -p $RETAIL_HOME/stage
```
6. Download the patch to the staging directory and unzip it.
7. Execute orpatch to analyze the patch.

```
orpatch analyze -s $RETAIL_HOME/stage/patch123456
```
8. Repeat the patch analysis on all servers with installations for this product environment.
9. Evaluate the list(s) of impacted files.

For more information on registering and analyzing customizations, please see the Customization section later in this document.

Applying a Patch

Once the system is prepared for patching, ORPatch can be executed to apply the patch to the environment. The patch may need to be applied to multiple systems if it updates components that are installed on distributed servers.

To apply a patch, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```

3. Set the PATH environment variable to include the orpatch/bin directory
4. Set the DISPLAY environment variable if the patch contains Forms.

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```

```
export DISPLAY=localhost:10.0
```

Note: If the DISPLAY environment variable is not specified, the value from RETAIL_HOME/orpatch/config/env_info.cfg will be used.

5. Set the JAVA_HOME environment variable if the patch contains Java application files.

```
export JAVA_HOME=/u00/oretail/java_jdk
```

Note: If the JAVA_HOME environment variable is not specified, the value from RETAIL_HOME/orpatch/config/env_info.cfg will be used.

6. Create a staging directory to contain the patch, if it does not already exist.
7. Download the patch to the staging directory and unzip it.
8. Review the README.txt included with the patch. If manual steps are specified in the patch, execute those steps at the appropriate time.
9. Shutdown applications.
10. Execute ORPatch to apply the patch.
11. After ORPatch completes, review the log files in \$RETAIL_HOME/orpatch/logs.
12. Repeat the patch application on all servers with installations for this product environment.
13. Restart applications.

```
mkdir -p $RETAIL_HOME/stage
```

```
orpatch apply -s $RETAIL_HOME/stage/patch123456
```

Restarting ORPatch

If ORPatch is interrupted while applying a patch, or exits with an error, it saves a record of completed work in a restart state file in \$RETAIL_HOME/orpatch/logs. Investigate and resolve the problem that caused the failure, then restart ORPatch.

By default when ORPatch is started again, it will restart the patch process close to where it left off. If the patch process should **not** be restarted, add '-new' to the command-line of ORPatch.

Please note that starting a new patch session without completing the prior patch may have serious impacts that result in a patch not being applied correctly. For example, if a patch contains database updates and batch file changes and ORPatch is aborted during the load of database objects, abandoning the patch session will leave batch without the latest changes compiled in the installation.

Listing the Patch Inventory

After a patch is successfully applied by ORPatch the patch inventory in `$RETAIL_HOME/orphatch/inventory` is updated with a record that the patch was applied. This inventory contains a record of the patches applied, the dates they were applied, the patch type and products impacted.

To list the patch inventory, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the `RETAIL_HOME` environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```

3. Set the `PATH` environment variable to include the `orphatch/bin` directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```

4. Execute `orphatch` to list the inventory.

```
orphatch lsinventory
```

Exporting Environment Metadata

ORPatch functionality is driven based on additional metadata that is stored in the environment to define what version of files are applied to the environment, and which database scripts have been applied to database schemas. This environment metadata is used to analyze the impact of patches to environments and controls what actions are taken during a patch. The metadata is stored in several locations depending on the type of information it tracks and in some cases it may be desirable to extract the metadata for analysis outside of ORPatch. For example, Oracle Support could ask for the metadata to be uploaded to assist them in triaging an application problem.

ORPatch provides a capability to export all of the metadata in an environment into a single directory and to automatically create a zip file of that content for upload or transfer to another system. The exact metadata collected from the environment depends on the products installed in the `RETAIL_HOME`.

ORPatch metadata exported:

Installed Product Component	Exported Metadata	Description
Any	<code>orphatch/config/env_info.cfg</code> <code>orphatch/config/custom_hooks.cfg</code> ORPatch inventory files	ORPatch configuration and settings
Any	All <code>env_manifest.csv</code> and <code>deleted_env_manifest.csv</code> files	Environment manifest files detailing product files installed, versions, customized flags and which patch provided the file
Database Schemas	DBMANIFEST table contents	Database manifest information detailing which database scripts were run, what version and when they were executed
Java Applications	All files from <code>javaapp_<product>/config</code> except jar files	Environment-specific product configuration files generated during installation

Installed Product Component	Exported Metadata	Description
Java Applications	Combined export of all META-INF/env_manifest.csv files from all product ear files	Jar manifest information detailing files, versions, customized flags and which patch provided the file
Java Applications	orpatch/config/javaapp_<product>/ant.deploy.properties	Environment properties file created during product installation and used during application deployment
Java Applications	<weblogic_home>/server/lib/weblogic.policy	WebLogic server java security manager policy file
RMS Batch	orpatch/config/rmsbatch_profile	Batch compilation shell profile
RMS Forms	orpatch/config/rmsforms_profile	Forms compilation shell profile
RWMS Forms	orpatch/cofngi/rwsmforms_profile	Forms compilation shell profile

Exports of environment metadata are always done to the \$RETAIL_HOME/support directory. When exporting metadata, you must specify the `-exname` argument and define the name that should be given to the export. The name is used for the directory within \$RETAIL_HOME/support and for the name of the zip file.

To extract an environment's metadata, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Execute orpatch to export the metadata.

```
orpatch exportmetadata -exname test_env
```

This example would export all metadata from the environment to the \$RETAIL_HOME/support/test_env directory. A zip file of the metadata would be created in \$RETAIL_HOME/support/test_env.zip.

Note: The \$RETAIL_HOME/support/<name> directory should be empty or not exist prior to running `exportmetadata` in order to ensure accurate results.

Comparing Environment Metadata

Once metadata has been exported from an environment, it can be used to compare the environment manifest metadata of two environments. ORPatch provides a capability to compare metadata of the current environment with the exported metadata of another environment. Note that even though there are many types of metadata exported by ORPatch, only environment manifest metadata is evaluated during comparisons. Metadata comparison happens in four phases: product comparison, patch comparison, ORPatch action comparison, and module-level comparison.

Product comparison compares the products installed in one environment with the products installed in another environment. Patch comparison compares the patches applied in one environment with the patches applied in another environment, for common products. This provides the most summarized view of how environments

differ. Patches which only apply to products on one environment are not included in the comparison.

Since each patch may impact many files, the comparison then moves on to more detailed analysis. The third phase of comparison is to compare the enabled ORPatch actions between environments. These actions roughly correspond to the installed 'components' of a product. For example, one environment may have database and forms components installed while another has only forms. Action comparison identifies components that are different between environments. The final phase of comparison is at the module level for actions that are common between environments. Modules which exist only on one environment, or exist on both environments with different revisions, or which are flagged as customized are reported during the comparison.

Differences between environment metadata are reported in a summarized fashion during the ORPatch execution. Details of the comparison results are saved in `$RETAIL_HOME/orphatch/logs/detail_logs/compare/details`. One CSV file is created for each phase of comparison: `product_details.csv`, `patch_details.csv`, `action_details.csv` and `module_details.csv`.

In order to be compared by ORPatch, exported metadata must be placed in the `$RETAIL_HOME/support` directory. The metadata should exist in the same structure that it was originally exported in. For example, if the metadata was exported to `$RETAIL_HOME/support/test_env` on another system, it should be placed in `$RETAIL_HOME/support/test_env` on this system.

When reporting differences between two environments, ORPatch uses names to refer to the environments. These names are defined as part of the `diffmetadata` command. The `-expname` parameter, which defines the directory containing the metadata, is also used as the name when referring to the exported metadata. The `-srcname` parameter defines the name to use when referring to the current environment. As an example, if you had exported the 'test' environment's metadata and copied it to the 'dev' environment's `$RETAIL_HOME/support/test_env` directory, you could run "`orphatch diffmetadata -expname test_env -srcname dev_env`". The detail and summary output would then refer to things that exist on dev but not test, revisions in the test environment versus revisions in the dev environment, etc.

ORPatch will automatically export the environment's current metadata to `$RETAIL_HOME/support/compare` prior to starting the metadata comparison.

To compare two environment's metadata, perform the following steps:

1. Export the metadata from another environment using `orphatch exportmetadata`.
2. Transfer the metadata zip from the other system to `$RETAIL_HOME/support`.
3. Log in as the UNIX user that owns the product installation.
4. Set the `RETAIL_HOME` environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/dev
```

5. Set the `PATH` environment variable to include the `orphatch/bin` directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```

6. Unzip the metadata zip file.

```
unzip test_env.zip
```

7. Execute `orphatch` to compare the metadata

```
orphatch diffmetadata -expname test_env -srcname dev_env
```

This example would compare the current environment against the metadata extracted in `$RETAIL_HOME/support/test_env` directory.

Note: The \$RETAIL_HOME/support/compare directory will be automatically removed before environment metadata is exported at the start of the comparison.

Reverting a Patch

In general it is best to either completely apply a patch, or restore the entire environment from the backup taken before starting the patch. It is important to test patches in test or staging environments before applying to production. In the event of problems, Oracle Retail recommends restoring the environment from backup if a patch is not successful.

Note: Reverting patches in an integrated environment can be extremely complex and there is no fully automated way to revert all changes made by a patch. Restoring the environment from a backup is the recommended method to remove patches.

It is, however, possible to revert small patches using the backups taken by ORPatch during a patch. This will restore only the files modified, and it is still necessary to restore the database if any changes were made to it.

Note: Reverting a patch reverts only the files modified by the patch, and does not modify the database, or recompile forms or batch files after the change.

When multiple patches have been applied to an environment, reverting any patches other than the most recently applied patch is strongly discouraged as this will lead to incompatible or inconsistent versions of modules applied to the environment. If multiple patches are going to be applied sequentially it is recommended to first merge the patches into a single patch that can be applied or reverted in a single operation.

To revert a patch, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Identify the backup directory in \$RETAIL_HOME/backups that contains the backup from the patch you want to restore.
 - The backup directory will contain a patch_info.cfg file which contains the name of the patch the backup is from.
 - It is possible to have two directories for the same patch, if ORPatch was updated during the patch. It is not possible to revert the updates to ORPatch. Select the backup directory that does not contain orpatch files.
 - If it is not clear which backup directory to use, restore the environment from backup
5. Execute orpatch to revert the environment using the contents of the backup directory

```
orpatch revert -s $RETAIL_HOME/backups/backup-11232013-152059
```
6. Restore the database from backup if the patch made database changes
7. Use the orcompile script to recompile forms if the patch included RMS or RWMS forms files

```
orcompile -a RMS -t FORMS
orcompile -a RWMS -t FORMS
```

8. Use the orcompile script to recompile batch if the patch included RMS batch files
orcompile -a RMS -t BATCH

9. Use the ordeploy script to redeploy the appropriate Java applications if the patch included Java files

```
ordeploy -a RPM -t JAVA
ordeploy -a REIM -t JAVA
ordeploy -a ALLOC -t JAVA
ordeploy -a RESA -t JAVA
```

Merging Patches

When patches are applied individually some ORPatch tasks such as compiling forms and batch files or deploying Java archives are performed separately for each patch. This can be time-consuming. An alternative is to use the ORMerge utility to combine several patches into a single patch, reducing application downtime by eliminating tasks that would otherwise be performed multiple times. Patches merged with ORMerge are applied with ORPatch after the merge patch is created.

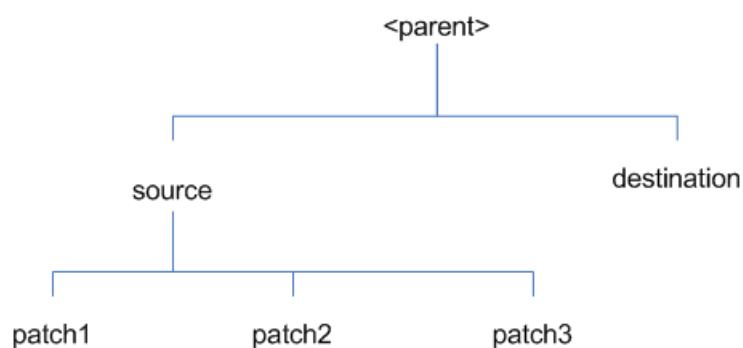
Source and Destination Directories

ORMerge uses source and destination areas in order to merge patch files. The source area is a single directory that contains the extracted patches to merge. The destination area is the location where the merged patch will be created. If a file exists in one or more source patches, only the highest revision will be copied to the merged patch.

The source and destination directories should exist under the same parent directory. That is, both the source and destination directories should be subdirectories of a single top-level directory.

The source directory must have all patches to be merged as immediate child directories. For example if three patches need to be merged the directory structure would look like this:

Source and Destination Directory Example



In the example above, the manifest.csv and patch_info.cfg files for each patch to be merged must exist in source/patch1, source/patch2, and source/patch3.

ORMerge Command-line Arguments

Argument	Required	Description
-s	Yes	Path to source directory containing patches to merge
-d	Yes	Path to destination directory that will contain merged patch
-name	No	The name to give the merged patch. If not specified, a name will be generated. When the merged patch is applied to a system, this name will appear in the Oracle Retail patch inventory.
-inplace	No	Used only when applying a patch to installation files prior to the first installation. See "Patching prior to the first install" in the Troubleshooting section later, for more information.

Running the ORMerge Utility

To merge patches, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```
4. Create a staging directory to contain the patches.

```
mkdir -p $RETAIL_HOME/stage/merge/src
```
5. Download the patches to the staging directory and unzip them so that each patch is in a separate subdirectory.
6. Review the README.txt included with each patch to identify additional manual steps that may be required. If manual steps are specified in any patch, execute them at the appropriate time when applying the merged patch.
7. Create a destination directory to contain the merged patches.

```
mkdir -p $RETAIL_HOME/stage/merge/dest
```
8. Execute ORMerge to merge the patches.

```
ormerge -s $RETAIL_HOME/stage/merge/src -d $RETAIL_HOME/stage/merge/dest -name merged_patch
```

The merged patch can now be applied as a single patch to the product installation using ORPatch.

Compiling Application Components

In some cases it may be desirable to recompile RMS Forms, RWMS Forms or RMS Batch outside of a product patch. The ORCompile utility is designed to make this easy and remove the need to manually execute 'make' or 'frmcmp' commands which can be error-prone. ORCompile leverages ORPatch functions to ensure that it compiles forms and batch exactly the same way as ORPatch. In addition ORCompile offers an option to compile invalid database objects using ORPatch logic.

ORCompile takes two required command line arguments each of which take an option. Arguments and options can be specified in upper or lower case.

ORCompile Command Line Arguments

Argument	Description
-a <app>	The application to compile.
-t <type>	The type of application objects to compile

ORCompile Argument Options

Application	Type	Description
RMS	BATCH	Compile RMS Batch programs
RMS	FORMS	Compile RMS Forms
RWMS	FORMS	Compile RWMS Forms
RMS	DB	Compile invalid database objects in the primary RMS schema
RMS	DB-ASYNC	Compile invalid database objects in the RMS_ASYNC_USER schema
ALLOC	DB-ALC	Compile invalid database objects in the Allocations user schema
ALLOC	DB-RMS	Compile invalid database objects in the RMS schema
REIM	DB	Compile invalid database objects in the RMS schema
RME	DB	Compile invalid database objects in the RME schema
ASO	DB	Compile invalid database objects in the ASO schema
RI	DB-DM	Compile invalid database objects in the RI DM schema
RI	DB-RIBATCH	Compile invalid database objects in the RI batch schema
RI	DB-RMSBATCH	Compile invalid database objects in the RI RMS batch schema
RI	DB-FEDM	Compile invalid database objects in the RI front-end schema
RDE	DB-DM	Compile invalid database objects in the RDE DM schema
RDE	DB-RDEBATCH	Compile invalid database objects in the RDE batch schema
RDE	DB-RMSBATCH	Compile invalid database objects in the RDE RMS batch schema

Note: Compiling RMS type DB, ReIM type DB, and Allocation type DB-RMS, are all identical as they attempt to compile all invalid objects residing in the RMS schema.

Running the ORCompile utility

To compile files, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orpatch/bin:$PATH
```

- Execute orcompile to compile the desired type of files.

```
orcompile -a <app> -t <type>
```

ORCompile Examples

Compile RMS Batch.

```
orcompile -a RMS -t BATCH
```

Compile RWMS Forms.

```
orcompile -a RWMS -t FORMS
```

Compile invalid objects in the RA DM schema.

```
orcompile -a RI -t DB-DM
```

Compile invalid objects in the RMS owning schema.

```
orcompile -a RMS -t DB
```

Deploying Application Components

In some cases it may be desirable to redeploy Java applications outside of a product patch. For example, when troubleshooting a problem, or verifying the operation of the application with different WebLogic settings. Another situation might include wanting to deploy the application using the same settings, but without customizations to isolate behavior that could be related to customized functionality.

The ordeploy utility is designed to make this easy and remove the need to re-execute the entire product installer when no configuration needs to change. ORDeploy leverages Oracle Retail Patch Assistant functions to ensure that it deploys applications exactly the same way as ORPatch. In addition ORDeploy offers an option to include or not include custom Java files, to ease troubleshooting.

ORDeploy takes two required command line arguments each of which take an option. Arguments and options can be specified in upper or lower case.

ORDeploy Command Line Arguments

Argument	Description
-a <app>	The application to deploy.
-t <type>	The type of application objects to deploy

ORDeploy Argument Options

Application	Type	Description
ALLOC	JAVA	Deploy the Allocations Java application and Java batch files, including any custom Java files.
ALLOC	JAVANOCUSTOM	Deploy the Allocations Java application and Java batch files, NOT including any custom Java files.
REIM	JAVA	Deploy the REIM Java application and Java batch files, including any custom Java files.
REIM	JAVANOCUSTOM	Deploy the REIM Java application and Java batch files, NOT including any custom Java files.

Application	Type	Description
RESA	JAVA	Deploy the RESA Java application, including any custom Java files.
RESA	JAVANOCUSTOM	Deploy the RESA Java application, NOT including any custom Java files.
RPM	JAVA	Deploy the RPM Java application and Java batch files, including any custom Java files.
RPM	JAVANOCUSTOM	Deploy the RPM Java application and Java batch files, NOT including any custom Java files.

Running the ORDeploy utility

To deploy Java applications, perform the following steps:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. Execute ORDeploy to deploy the desired Java application.

```
ordeploy -a <app> -t <type>
```

ORDeploy Examples

Deploy RPM.

```
ordeploy -a RPM -t JAVA
```

Deploy ReIM without including Java customizations.

```
ordeploy -a REIM -t JAVANOCUSTOM
```

Maintenance Considerations

The additional information stored within the RETAIL_HOME and within database schemas adds some considerations when performing maintenance on your environment.

Database Password Changes

Oracle wallets are used to protect the password credentials for connecting to database schemas. This includes all database schemas used during an install. If the password for any of these users is changed the wallet's entry must be updated.

The wallet location is configurable but by default is in the following locations:

Location	Installation Type
\$RETAIL_HOME/orphatch/rms_wallet	RMS Database RMS Batch
\$RETAIL_HOME/orphatch/rms_wallet_app	RMS Forms
\$RETAIL_HOME/orphatch/rwms_wallet	RWMS Database
\$RETAIL_HOME/orphatch/rwms_wallet_app	RWMS Forms
\$RETAIL_HOME/orphatch/oraso_wallet	ASO Database

Location	Installation Type
\$RETAIL_HOME/orpatch/orme_wallet	RME Database
\$RETAIL_HOME/orpatch/ra_wallet	RI (Previously RA) Database
\$RETAIL_HOME/orpatch/rde_wallet	RDE Database

The wallet alias for each schema will be <username>_<dbname>. Standard mkstore commands can be used to update the password.

For example:

```
mkstore -wrl $RETAIL_HOME/orpatch/rms_wallet -modifyCredential rms_rmsdb rms01
rmspassword
```

This command will update the password for the RMS01 user to 'rmspassword' in the alias 'rms_rmsdb'.

The Oracle wallets are required to be present when executing ORPatch. Removing them will prevent you from being able to run ORPatch successfully. In addition the Oracle wallet location is referenced in the RMS batch.profile, and in the default RMS and RWMS Forms URL configuration, so removing them will require reconfiguration of batch and forms. If batch and forms were reconfigured after installation to use other wallet files, it is possible to backup and remove the wallets, then restore them when running ORPatch.

WebLogic Password Changes

Java wallets are used to protect the password credentials used when deploying Java products. This includes the WebLogic administrator credentials, LDAP connection credentials, batch user credentials and any other credentials used during an install. If the password for any of these users is changed the wallet's entry must be updated, or the Java product installation can be run again.

The wallet location is in the following locations:

Location	Installation Type
\$RETAIL_HOME/orpatch/config/javapp_rpm	RPM Java
\$RETAIL_HOME/orpatch/config/javapp_reim	ReIM Java
\$RETAIL_HOME/orpatch/config/javapp_alloc	Allocation Java
\$RETAIL_HOME/orpatch/config/javapp_resa	RESA Java
\$RETAIL_HOME/orpatch/config/javaapp_rasrm	ORAAC (Previously RASRM) Java

The wallet aliases will be stored in the retail_installer partition. The names of the aliases will vary depending on what was entered during initial product installation.

The dump_credentials.sh script can be used to list the aliases in the wallet.

For example:

```
cd $RETAIL_HOME/orpatch/deploy/retail-public-security-api/bin
./dump_credentials.sh $RETAIL_HOME/orpatch/config/javapp_alloc
```

```
Application level key partition name:retail_installer
```

```
User Name Alias:dsallocAlias User Name:rms01app
```

```
User Name Alias:BATCH-ALIAS User Name:SYSTEM_ADMINISTRATOR
```

```
User Name Alias:wlsAlias User Name:weblogic
```

The easiest way to update the credential information is to re-run the Java product installer. If you need to manually update the password for a credential, the `save_credential.sh` script can be used.

For example:

```
cd $RETAIL_HOME/orpatch/deploy/retail-public-security-api/bin
./save_credential.sh -l $RETAIL_HOME/orpatch/config/javapp_alloc -p
retail_installer -a wlsAlias -u weblogic
```

This command will prompt for the new password twice and update the alias `wlsAlias`, username `weblogic` with the new password.

Infrastructure Directory Changes

The `RETAIL_HOME/orpatch/config/env_info.cfg` file contains the path to the database `ORACLE_HOME` on database or RMS Batch installations, to the WebLogic Forms and Reports `ORACLE_HOME` and `ORACLE_INSTANCE` on RMS or RWMS Forms installations, and to the `WEBLOGIC_DOMAIN_HOME`, `WL_HOME` and `MW_HOME` on Java product installations. If these paths change, the related configuration variables in the `env_info.cfg` file must be updated.

DBManifest Table

The table `dbmanifest` within Oracle Retail database schemas is used to track the database scripts which have been applied to the schema. It is critical not to drop or truncate this table. Without it, `ORPatch` will attempt to re-run scripts against the database which have already been applied which can destroy a working environment. Similarly, if copying a schema from one database to another database, ensure that the `dbmanifest` table is preserved during the copy.

RETAIL_HOME relationship to Database and Application Server

The `RETAIL_HOME` associated with an Oracle Retail product installation is critical due to the additional metadata and historical information contained within it. If a database or application installation is moved or copied, the `RETAIL_HOME` related to it should be copied or moved at the same time.

Jar Signing Configuration Maintenance

The RPM product installation includes an option to configure a code signing certificate so that jar files modified during installation or patching are automatically re-signed. This configuration is optional, but recommended. If it is configured, the code signing keystore is copied during installation to `$RETAIL_HOME/orpatch/config/jarsign/orpkeystore.jks`. The keystore password and private key password are stored in a Java wallet in the `$RETAIL_HOME/orpatch/config/jarsign` directory. The credentials are stored in a wallet partition called `orpatch`:

Alias	Username	Description
storepass	discard	Password for the keystore
keypass	discard	Password for the private key

The keystore file and passwords can be updated using the product installer. This is the recommended way to update the signing configuration.

If only the credentials need to be updated, the `sign_jar.sh` script can be used.

5. Log in as the UNIX user that owns the product installation.
6. Set the `RETAIL_HOME` environment variable to the top-level directory of your installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
7. Change directories to the location of `sign_jar.sh`

```
cd $RETAIL_HOME/orpatch/deploy/bin
```
8. Execute `sign_jar.sh`

```
sign_jar.sh changepwd
```
9. When prompted, enter the new keystore password
10. When prompted, enter the new private key password

Customization

Patching Considerations with Customized Files and Objects

In general, the additional capabilities provided by the ORPatch should make it easier to evaluate the potential impacts of patches to your customizations of Oracle Retail products. However, the additional metadata maintained by the Oracle Retail patching utilities does add some considerations when making customizations.

General Guidelines

It is always preferred to customize applications by extension rather than by direct modification. For example, adding new database objects and forms rather than modifying existing Oracle Retail objects and forms. You can also leverage built-in extension points such as User Defined Attributes, the Custom Flexible Attribute Solution, or seeded customization points in ADF Applications.

It is strongly discouraged to directly modify Oracle Retail database objects, especially tables, as your changes may be lost during patching or may conflict with future updates. When adding or modifying database objects, Oracle Retail recommends that all objects be added with scripts to ensure that they can be rebuilt if necessary after a patch.

Custom Database Objects

When you create new database objects, Oracle Retail recommends placing them in an Oracle database schema specifically for your customizations. You must use synonyms and grants to allow the Oracle Retail product schema owner and other users to access your objects, and use synonyms and grants to allow your customizations to access Oracle Retail objects. A separate schema will ensure that your customizations are segregated from base Oracle Retail code.

ORPatch expects that there will be no invalid objects in the database schemas it manages after a patch is applied. For this reason adding extra objects to the product schema could result in failures to apply patches as changes to base objects may cause custom objects to go invalid until they are updated. In this situation, manually update the custom objects so that they compile, and restart the patch.

Custom Forms

When creating new custom forms, Oracle Retail recommends placing them in a separate directory specifically for your customizations. This directory should be added to the FORMS_PATH of your RMS or RWMS Forms URL configuration to allow the forms to be found by the Forms Server. This will ensure that your customizations are segregated from base Oracle Retail code. If you choose to place customizations in the Forms bin directory, then your custom forms will need to be recopied each time Forms are fully recompiled.

ADF Application Customization

Oracle Retail ADF-based applications such as Allocation and ReSA can be customized using a process called 'seeded customization'. The customization process involves using JDeveloper in Customizer mode to create changes to product configurations, and then building a MAR archive containing the changes. The generated MAR is deployed to the MDS repository used by the application and applied to the application at runtime. These types of customizations are handled outside of ORPatch and are not reported during patch analysis or tracked by the custom file registration utility. More information can be found in the respective product customization guides.

Custom Compiled Java Code

When customizing Oracle Retail Java-based products such as RPM and ReIM via product source code, ORPatch supports automatically adding compiled customizations into the application ear file prior to deployment. This allows customizations to be applied to the application without directly modifying the base product ear, enabling customizations and defect hotfixes to co-exist when they do not change the same file or a dependent file. See the later “Custom Compiled Java Code” section for additional information and considerations.

Analyze Patches when Customizations are Present

Whenever you have customized a product by directly modifying Oracle Retail files or database objects, it is important to ensure you analyze each the files that will be updated by a patch before applying the patch. This will allow you to identify any customized files which may be overwritten by the patch and either merge your customization with the new version of the file, or re-apply the customization after applying the patch.

Manifest Updates

If you choose to customize Oracle Retail files directly, it is extremely important **not** to update the revision number contained in the env_manifest.csv. This could cause future updates to the file to be skipped, invalidating later patch applications as only a partial patch would be applied. The customized revision number for modified files will need to be tracked separately.

Registering Customized Files

The ORPatch contains utilities and functionality to allow tracking of files that have been customized through direct modification. This process is referred to as ‘registering’ a customized file. Registration only works for files which are shipped by Oracle Retail. It is not possible to register new files created in the environment as part of extensions or customizations.

When patches are analyzed with ORPatch, special reporting is provided if any registered files would be updated or deleted by the patch. Customized files impacted by the patch are listed at the end of the analysis report from ORPatch. The detail files generated during the analyze will contain a column called ‘customized’ which will have a Y for any files which were registered as customized. This allows easier identification of customizations which will be overwritten by a patch.

All files delivered by Oracle Retail are considered ‘base’ and so when they are applied to an environment any registrations of those files as customized will revert back to un-customized. **Each time a patch overwrites customized files, you must re-register the files as customized once you have applied customizations.**

To register customized files, use the \$RETAIL_HOME/orpatch/bin/orcustomreg script. The orcustomreg script operates in one of two modes: registration and list.

- Registration mode registers or unregisters one or more files as customized.
- List mode lists all files in the environment that are registered as customized.

Command Line Arguments for Registration Mode

Argument	Description
-f <file>	Adds <file> to the list of files that will be registered. Can be specified more than once.

Argument	Description
-bulk <file>	Specifies a file to read, containing one filename per line. All filenames listed inside <file> will be registered.
-register	Files specified with -f or -bulk will be registered as 'customized'
-unregister	Files specified with -f or -bulk will be registered as 'base'

Notes:

- At least one of -f or -bulk is required.
 - If neither -register nor -unregister is specified, the default is '-register'.
 - File names specified with -f must either be fully-qualified or be relative to RETAIL_HOME. The same is true for filenames specified within a -bulk file.
-
-

Command Line arguments for list mode

Argument	Description
-list	List all files in the environment registered as customized

Running the orcustomreg Script

Perform the following procedure to run the orcustomreg script:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. Execute orcustomreg script to register the desired file(s).

```
orcustomreg -register -f <file>
```

Examples of using the orcustomreg Script

Register \$RETAIL_HOME/dbsql_rms/Cross_Pillar/control_scripts/source/oga.sql as customized.

```
orcustomreg -f dbsql_rms/Cross_Pillar/control_scripts/source/oga.sql
```

Unregister customizations for

```
$RETAIL_HOME/dbsql_rwms/Triggers/Source/TR_WAVE.trg
```

```
orcustomreg -unregister -f $RETAIL_HOME/dbsql_rwms/Triggers/Source/TR_WAVE.trg
```

Bulk register several files as customized.

```
echo "$RETAIL_HOME/oracle/proc/src/mrt.pc" > custom.txt
echo "$RETAIL_HOME/oracle/proc/src/saldly.pc" >> custom.txt
echo "$RETAIL_HOME/oracle/proc/src/ccprg.pc" >> custom.txt
orcustomreg -bulk custom.txt
```

List all files registered as customized.

```
orcustomreg -list
```

Custom Compiled Java Code

When customizing Oracle Retail Java-based products such as RPM and ReIM via product source code, ORPatch supports automatically adding compiled customizations into the application ear file prior to deployment. This allows customizations to be applied to the application without directly modifying the base product ear, enabling customizations and defect hotfixes to co-exist when they do not change the same file or a dependent file.

This functionality is enabled by creating a directory called `$RETAIL_HOME/javaapp_<app>/custom`, where `<app>` is the application the customizations apply to. Files stored within this directory will be combined with the base product ear files before the application is deployed to WebLogic. ORPatch will attempt to consider customizations stored within the 'custom' directory during patch analysis by triggering more detailed ear file change analysis to assist with identifying which customizations might be impacted by changes in the patches.

Note: It is not possible, nor necessary, to register compiled Java customizations with the `orcustomreg` tool.

As with other customization techniques for other technologies, Oracle Retail recommends making Java customizations in new files as much as possible, versus overwriting base product or configuration files. In the past it was necessary to build complete replacement product ear files, but this method of customization is no longer required nor recommended. Replacement ear and jar files will not contain the `META-INF/env_manifest.csv` files which are required in order to be able to apply incremental patches. Instead, compile the specific Java classes being customized and place them along with any custom configuration files in `$RETAIL_HOME/javaapp_<app>/custom`.

Building Deployable ear files

When constructing the product ear file to deploy to WebLogic, ORPatch applies changes to the ear file in a specific order, with files from later steps overwriting files in earlier steps. The resulting ear is stored in `$RETAIL_HOME/javaapp_<app>/deploy`, and then deployed to WebLogic.

Sequence for ORPatch Java Product ear file updates

Order	File Type	Location
1	Base product ear	<code>\$RETAIL_HOME/javaapp_<app>/base</code>
2	Updated configuration files	<code>\$RETAIL_HOME/javaapp_<app>/config</code>
3	Oracle Retail-supplied hotfixes	<code>\$RETAIL_HOME/javaapp_<app>/internal</code>
4	Compiled customizations	<code>\$RETAIL_HOME/javaapp_<app>/custom</code>

Merging Custom Files

When merging files from the custom directory with the product ear, ORPatch uses the directory path of the files within custom to calculate where the file should be stored within the ear. This allows arbitrary nesting of files, even when placing files within jars stored in jars, stored within the ear. The following examples below use RPM, but apply to adding compiled customizations to any Java-based product.

Custom directory location and product ear location Examples

File path within javaapp_<app>/custom/	Final Ear File Location
rpm.ear/company/ui/MyCustom.class	In rpm.ear: /company/ui/MyCustom.class
rpm.ear/rpm.jar/company/bc/MyCustom2.class	In rpm.ear: In rpm.jar: /company/bc/MyCustom2.class
rpm.ear/lib/ourcustomlibs.jar	In rpm.ear /lib/ourcustomlibs.jar
rpm.ear/WebLaunchServlet.war/lib/ rpm.jar/company/bc/MyCustom2.class	In rpm.ear: In WebLaunchServlet.war: In lib/rpm.jar: /company/bc/MyCustom2.class

Analyzing patches when customizations are present

When analyzing a patch which contains a base product ear and the custom directory contains files, ORPatch will automatically trigger a more detailed analysis of the changes coming in a patch. This includes calculating what files inside the product ear have been added, removed or updated and which files appear to be customized based on the contents of the 'custom' directory. The detailed results of the ear file comparison during patch analysis will be saved in javaapp_<app>_archive_compare_details.csv. Any custom files which appeared to be impacted by the patch are saved in javapp_<app>_archive_custom_impacts.csv. Both files will be in the \$RETAIL_HOME/orpatch/logs/detail_logs/analyze/details directory.

Note: This detailed analysis is not available when analyzing individual hotfixes, so special care must be taken when applying hotfixes to a customized product installation, to ensure there are no conflicts between customizations and hotfix changes.

Customizations and cumulative patches

By default, when applying a cumulative patch, ORPatch will not include customizations in the deployed product ear, even if they are present in the appropriate directory. This allows verification that the application is functioning properly using base code, before applying customizations. After verifying the initial deployment, use ORDeploy with the "-t JAVA" option to construct and deploy the product ear including customizations.

If customizations need to be removed outside of a patch, use ORDeploy with the "-t JAVANOCUSTOM" option to create and deploy an ear containing only Oracle Retail code. To force ORPatch to include customizations in the deployed ear even when applying a cumulative patch, set JAVAAPP_<app>_INCLUDE_CUSTOM=Y in the \$RETAIL_HOME/orpatch/config/env_info.cfg file.

Changing configuration files

It is possible to directly change product configuration files in \$RETAIL_HOME/javaapp_<app>/config. These updates can be deployed to the environment using the ORDeploy utility. However, the 'config' directory is completely

recreated each time the product installer is used. This means that modifications will be lost and must be manually reapplied after each installer run. It is recommended to make configuration changes via the installer where possible, and retain the `ant.install.properties` file for use in later installer sessions.

Extending Oracle Retail Patch Assistant with Custom Hooks

The default ORPatch actions and processing logic is sufficient to install and patch the base Oracle Retail product code. However there may be situations where custom processing is desired during patching activities such as executing a shell script prior to the start of patching, or running a SQL script at the end of the patch.

ORPatch supports extensions in the form of custom hooks. These hooks allow external scripts to be run at specific points during ORPatch processing.

ORPatch Processing

Action

ORPatch supports a variety of 'actions' which define the steps necessary to apply updates to a particular area of the Oracle Retail application. Each action is generally specific to updates to a single technology or logical component of the environment. For example, one action might handle making updates to the RMS database schema, while a separate action is responsible for compiling RWMS forms, and a different action deploys the RPM Java application. These actions are enabled and disabled within the environment configuration file, allowing ORPatch to determine what types of changes to apply to each product installation.

ORPatch Actions

Order	Action Name	Description
1	DBSQL_RMS	Loads RMS and RPM database objects into the primary RMS schema
2	DBSQL_RMSASYNC	Loads database objects into the RMS_ASYNC_USER schema
3	DBSQL_RAF	Loads Retail Application Framework database objects into the RMS schema
4	DBSQL_REIM	Loads ReIM database objects into the RMS schema
5	DBSQL_ALCRMS	Loads Allocation database objects into the RMS schema
6	DBSQL_ALLOC	Loads Allocation database objects into the Allocation user schema
7	DBSQL_RMSDEMO	Used to create demo data in the RMS schema if demo data was selected during initial installation
8	DBSQL_RMSDAS	Loads database objects into the RMS Data Access Schema
9	RMSBATCH	Compiles RMS Batch
10	ORAFORMS_RMS	Compiles RMS Forms, copies RMS reports to \$RETAIL_HOME
11	RMSRETLSCRIPTS	Copies Oracle Retail Extract and Load scripts for RMS
12	RMSDCSCRIPTS	Copies Oracle Retail Merchandising System data conversion scripts

Order	Action Name	Description
13	DBSQL_RWMS	Loads database objects into the primary RWMS schema
14	DBSQL_RWMSADF	Loads database objects into the RWMS ADF user schema
15	DBSQL_RWMSUSER	Loads database objects into the RWMS user schema
16	ORAFORMS_RWMS	Compiles RWMS Forms, copies RWMS batch scripts and reports to \$RETAIL_HOME
17	JAVAAPP_RPM	Deploys the RPM Java application and batch scripts
18	JAVAAPP_REIM	Deploys the REIM Java application and batch scripts
19	JAVAAPP_ALLOC	Deploys the Allocation Java application and batch scripts
20	JAVAAPP_RESA	Deploys the ReSA Java application
21	JAVAAPP_RASRM	Deploys the ORAAC (previously called RASRM) Java application
22	DBSQL_RARMSBATCH	Loads database objects into the RMS Batch schema for RI (previously called RA)
23	DBSQL_RADM	Loads database objects into the RI (previously called RA) Data Mart schema
24	DBSQL_RAFEDM	Loads database objects into the RI (previously called RA) Front-end schema
25	DBSQL_RABATCH	Loads database objects into the RI (previously called RA) Batch schema
26	DBSQL_RDERMSBATCH	Loads database objects into the RMS Batch schema for RDE
27	DBSQL_RDEDM	Loads database objects into the RDE Data Mart schema
28	DBSQL_RDEBATCH	Loads database objects into the RDE Batch schema
29	DBSQL_RASECORE	Loads core database objects into the ORASE schema
30	DBSQL_RASEASO	Loads ASO database objects into the ORASE schema
31	DBSQL_RASECDT	Loads CDT database objects into the ORASE schema
32	DBSQL_RASECIS	Loads CIS database objects into the ORASE schema
33	DBSQL_RASEDT	Loads DT database objects into the ORASE schema
34	DBSQL_RASEMBA	Loads MBA database objects into the ORASE schema
35	RASECOREBATCH	Copies ORASE core batch scripts and libraries
36	RASEASOBATCH	Copies ORASE ASO batch scripts and libraries
37	RASECDTBATCH	Copies ORASE CDT batch scripts and libraries
38	RASECISBATCH	Copies ORASE CIS batch scripts and libraries
39	RASEDTBATCH	Copies ORASE DT batch scripts and libraries
40	RASEMBABATCH	Copies ORASE MBA batch scripts and libraries

Phase

ORPatch processes patches in phases. Each action relevant to a patch and host is provided an opportunity to process the patch for each phase. The standard phases which allow hooks are:

Restart Phase Number	Phase Name	Description
N/A	PRECHECK	Actions verify that their configuration appears complete and correct. This phase and the associated hooks will be run every time orpatch is executed, even if processing will be restarted in a later phase.
10	PREACTION	Actions do processing prior to when files are copied to the environment. Files are deleted during this phase.
20	COPYPATCH	Actions copy files included in a patch into the destination environment and the environment manifest is updated.
30	PATCHACTION	Actions take the more detailed steps necessary to apply the new files to the environment. For database actions in particular, this is the phase when new and updated sql files are loaded into the database.
40	POSTACTION	Actions do processing after files have been copied and PatchActions are completed. The Forms actions, for example, use this phase to compile the forms files as this must happen after database packages are loaded.
50	CLEANUP	Actions do any additional processing. Currently no actions implement activities in this phase.

Configuring Custom Hooks

Custom hooks are configured in a configuration file `RETAIL_HOME/orpatch/config/custom_hooks.cfg`. The configuration file is a simple text file where blank lines and lines starting with `#` are ignored and all other lines should define a custom hook.

To define a custom hook, a line is added to the file in the form:

```
<hook name>=<fully qualified script>
```

The hook name must be in upper case and is in the form:

```
<action name>_<phase name>_<sequence>
```

The action name is any action name understood by ORPatch. The phase name is one of the five phase names from the table above. The sequence is either 'START' or 'END'. Hooks defined with a sequence of 'START' are run before the action's phase is invoked. Hooks defined with a sequence of 'END' are run after the action's phase is invoked.

Multiple scripts can be associated with a single hook by separating the script names with a comma. If a hook name appears in the configuration file multiple times only the last entry will be used.

The script defined as a custom hook must be an executable shell script that does not take any arguments or inputs. The only environment variable that is guaranteed to be passed to the custom hook is `RETAIL_HOME`. The script must return 0 on success and non-zero on failure.

If an action is a DBSQL action (i.e. has a name like `DBSQL_`), the custom hook can optionally be a `.sql` file. In this case the SQL script will be run against the database schema that the DBSQL action normally executes against. The SQL script must not

generate any ORA- or SP2- errors on success. In order to be treated as a database script, the extension of the file defined as the custom hook must be .sql in lower-case. Any other extension will be treated as if it is a shell script. If you have database scripts with different extensions, they must be renamed or wrapped in a .sql script.

When using the PRECHECK phase and START sequence, please note that the custom hook will be executed prior to any verification of the configuration. Invalid configuration, such as invalid database username/password or a non-existent ORACLE_HOME, may cause the custom hook to fail depending on the actions it tries to take. However in these cases, the normal orpatch PRECHECK activities would likely have failed as well. All that is lost is the additional context that orpatch would have provided about what was incorrect about the configuration.

Restarting with Custom Hooks

If a custom hook fails, for example a shell script hook returns non-zero or a sql script generates an ORA- error in its output, the custom hook will be treated as failing. A failing custom hook causes ORPatch to immediately stop the patching session.

When ORPatch is restarted it always restarts with the same phase and action, including any START sequence custom hooks. If the START sequence custom hook fails, the action's phase is never executed. With an END sequence custom hook, the action's phase is re-executed when ORPatch is restarted and then the custom hook is re-executed.

When an action's phase is costly, for example the DBSQL_RMS action which does a lot of work, this can mean a lot of duplicate processing.

For this reason it is preferred to use START sequence custom hooks whenever possible. If necessary, use a START sequence hook on a later phase or a later action, rather than an END sequence custom hook.

Patch-level Custom Hooks

In addition to action-specific hooks, there are two patch-level hook points available. These hooks allow scripts to be run before any patching activities start and after all patching activities are completed. The hooks are defined in the same configuration file, with a special hook name.

To run a script before patching, define:

```
ORPATCH_PATCH_START=<fully qualified script>
```

To run a script after patching, define:

```
ORPATCH_PATCH_END=<fully qualified script>
```

These hooks only support executing shell scripts, database scripts must be wrapped in a shell script. It is also important to note that these hooks are run on every execution of ORPatch to apply a patch, even when restarting a patch application. If the START sequence patch-level hook returns a failure, patching is aborted. If the END sequence patch-level hook returns a failure, it is logged but ignored as all patching activities have already completed.

Please note that the ORPATCH_PATCH_START hook is executed prior to any verification of the configuration. Invalid configuration may cause the custom hook to fail depending on the actions it tries to take. However in these cases, the normal ORPatch activities would likely fail as well.

Example Custom Hook Definitions

A shell script that is executed prior to the Pre-Action phase of RMS Batch:

```
RMSBATCH_PREACTION_START=/u00/oretail/prepare_custom_header.sh
```

A shell script that is executed after RETL script files are copied into the RETAIL_HOME:

```
RETLSCRIPTS_COPYPATCH_END=/u00/oretail/copy_custom_files.sh
```

A SQL script that is executed against the RWMS owning schema at the start of the Clean-up Phase:

```
DBSQL_RWMS_CLEANUP_START=/dba/sql/recompile_synonyms.sql
```

Troubleshooting Patching

There is not a general method for determining the cause of a patching failure. It is important to ensure that patches are thoroughly tested in a test or staging system several times prior to attempting to apply the patch to a production system, particularly if the patch is a large cumulative patch. After the test application is successful, apply the patch to the production system.

ORPatch Log Files

ORPatch records extensive information about the activities during a patch to the log files in `RETAIL_HOME/orpatch/logs`. This includes a summary of the actions that are planned for a patch, information about all files that were updated by the patch, and detailed information about subsequent processing of those files. The ORPatch log files also contain timestamps to assist in correlating log entries with other logs.

Even more detailed logs are available in `RETAIL_HOME/orpatch/logs/detail_logs` for some activities such as forms compilation, invalid database object errors, and output from custom hooks. If the standard ORPatch log information is not sufficient, it might be helpful to check the detailed log if it exists.

Restarting ORPatch

The restart mechanism in ORPatch is designed to be safe in nearly any situation. In some cases to ensure this, a portion of work may be redone. If the failure was caused by an intermittent issue that has been resolved, restarting ORPatch may be sufficient to allow the patch to proceed.

Manual DBManifest Updates

A possible cause for database change script failures is that a database change was already made manually to the database. In this event, you may need to update the dbmanifest table to record that a specific script does not need to be run. Before doing this, it is extremely important to ensure that all statements contained in the script have been completed.

Use the `$RETAIL_HOME/orpatch/bin/ordbmreg` script to register database scripts in the dbmanifest table.

Command Line Arguments for ordbmreg

Argument	Description
<code>-f <file></code>	Adds <file> to the list of files that will be registered. Can be specified more than once.
<code>-bulk <file></code>	Specifies a file to read, containing one filename per line. All filenames listed inside <file> will be registered.
<code>-register</code>	Files specified with <code>-f</code> or <code>-bulk</code> will be registered in the dbmanifest table
<code>-unregister</code>	Files specified with <code>-f</code> or <code>-bulk</code> will be removed from the dbmanifest table

Notes:

- At least one of -f or -bulk is required.
 - If neither -register nor -unregister is specified, the default is '-register'.
 - File names specified with -f must either be fully-qualified or be relative to RETAIL_HOME. The same is true for filenames specified within a -bulk file.
 - Registering a file in the dbmanifest table will cause it to be completely skipped. Before doing so, ensure that all commands contained in it have been completed.
 - Removing a file from the dbmanifest table will cause it to be run again. This will fail if the commands in the script cannot be re-run. For example if they create a table that already exists.
-
-

Running the ordbmreg Script

Perform the following procedure to run the ordbmreg script:

1. Log in as the UNIX user that owns the product installation.
2. Set the RETAIL_HOME environment variable to the top-level directory of your product installation.

```
export RETAIL_HOME=/u00/oretail/15.0/tst
```
3. Set the PATH environment variable to include the orpatch/bin directory

```
export PATH=$RETAIL_HOME/orphatch/bin:$PATH
```
4. Execute ordbmreg script to register the desired file(s).

```
ordbmreg -register -f <file>
```

Examples of using the ordbmreg Script

Register

\$RETAIL_HOME/dbsql_rms/Cross_Pillar/db_change_scripts/source/000593_system_options.sql with the dbmanifest table.

```
ordbmreg -f
dbsql_rms/Cross_Pillar/db_change_scripts/source/000593_system_options.sql
```

Remove the dbmanifest row for

\$RETAIL_HOME/dbsql_radm/ra_db/radm/database_change_scripts/000035_s12733240_w_party_per_d.sql.

```
ordbmreg -unregister -f
$RETAIL_HOME/dbsql_radm/ra_db/radm/database_change_scripts/000035_s12733240_w_party_per_d.sql
```

Bulk register several files in the dbmanifest table.

```
echo "$RETAIL_HOME/dbsql_rwms/DBC/Source/000294_container.sql" > dbcs.txt
echo "$RETAIL_HOME/dbsql_rwms/DBC/Source/000457_drop_object.sql" >> dbcs.txt
ordbmreg -bulk dbcs.txt
```

Restarting after registration

Once the row has been added to the dbmanifest table, restart ORPatch and the script will be skipped. If the file is not skipped there are several possibilities:

- The script registered is not the failing script.
- The file type is not a type that is filtered by the dbmanifest. The only file types that skip files listed in the dbmanifest are:
 - Initial install DDL Files
 - Installation scripts that cannot be rerun
 - Database Change Scripts

Manual Restart State File Updates

Oracle Retail strongly discourages manually updating the ORPatch restart state files. Updating the file improperly could cause necessary steps in the patching process to be skipped or patches to be incorrectly recorded as applied.

DISPLAY Settings When Compiling Forms

When compiling RMS or RWMS forms, it is necessary to have a valid X-Windows Display. ORPatch allows this setting to come from one of two places:

- DISPLAY environment variable set before executing ORPatch
- or
- DISPLAY setting in RETAIL_HOME/orpatch/config/env_info.cfg

The DISPLAY variable in the environment overrides the env_info.cfg, if both are set. The destination X-Windows display must be accessible to the user running ORPatch, and for best compilation performance it should be on the network 'close' to the server where RMS Forms are installed and compiled. Using a local display or VNC display is preferred. Compiling forms across a Wide-Area Network will greatly increase the time required to apply patches to environments.

JAVA_HOME Setting

When working with Java application jar, ear or war files, it is necessary to have a valid JAVA_HOME setting. ORPatch allows this setting to come from one of two places:

- JAVA_HOME environment variable set before executing ORPatch
- or
- JAVA_HOME setting in RETAIL_HOME/orpatch/config/env_info.cfg

The JAVA_HOME variable in the environment overrides the env_info.cfg, if both are set. The specified Java home location must be accessible to the user running ORPatch and be a full Java Development Kit (JDK) installation. The JAVA_HOME must contain the jar utility and if automatic Jar file signing is configured, must also contain the keytool and jarsigner utilities.

Patching Prior to First Install

In some situations, it may be necessary to apply a patch to product installation files before the initial install. For example, if there is a defect with a script that would be run during the install and prevent proper installation. In this rare situation, it may be necessary to apply a patch to the installation files prior to starting installation.

Note: These steps should only be undertaken at the direction of Oracle Support.

Perform the following steps to patch installation files prior to starting an installation. The steps assume an RMS installation, but apply to any product supported by ORPatch:

1. Unzip the installation files to a staging area.

Note: The following steps assume the files are in
/media/oretail15.0

2. Locate the patch_info.cfg within the product media. The directory it resides in will be used for later steps.

```
find /media/oretail15.0/rms/installer -name patch_info.cfg
```

Output Example:

```
/media/oretail15.0/rms/installer/mom15/patch_info.cfg
```

3. Get the PATCH_NAME for the standard product installation. The patch name to use in subsequent steps will be the portion following the "=" sign.

```
grep "PATCH_NAME=" /media/oretail15.0/rms/installer/mom15/patch_info.cfg
```

Output Example:

```
PATCH_NAME=MOM_15_0_0_0
```

4. Create a directory that will contain the patch that must be applied, next to the directory with the product installation files.

Note: The following steps assume this directory is in
/media/patch.

5. Unzip the patch into the directory created in step 2.

Note: This should place the patch contents in
/media/patch/<patch num>.

6. Export RETAIL_HOME to point within the installation staging area.

```
export RETAIL_HOME=/media/oretail15.0/rms/installer/mom15/Build
```

7. Create a logs directory within the installation staging area

```
mkdir $RETAIL_HOME/orpatch/logs
```

8. Ensure the ORMerge shell script is executable.

```
chmod u+x $RETAIL_HOME/orpatch/bin/ormerge
```

9. Run ORMerge to apply the patch to the installation media, using a -name argument that is the same as what was found in step 3.

```
$RETAIL_HOME/orpatch/bin/ormerge -s /media/patch -d  
/media/oretail15.0/rms/installer/mom15 -name MOM_15_0_0_0 -inplace
```

Note: The -inplace argument is critical to ensure that the
patching replaces files in the mom15 directory.

10. Unset the RETAIL_HOME environment variable.

```
unset RETAIL_HOME
```

At this point, the installation files will have been updated with the newer versions of files contained within the patch. Log files for the merge will be in
/media/oretail15.0/rms/installer/mom15/Build/orpatch/logs.

Providing Metadata to Oracle Support

In some situations, it may be necessary to provide details of the metadata from an environment to Oracle support in order to assist with investigating a patching or application problem. ORPatch provides built-in functionality through the 'exportmetadata' action to extract and consolidate metadata information for uploading to

Oracle Support or for external analysis. For more information, see the ORPatch 'Exporting Environment Metadata' section.

Appendix: Oracle Database 12cR1 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='12.1.0.2'
*.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=''
*.plssql_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'
```

Appendix: Tablespace Creation Scripts

Non-Encrypted Tablespace Creation

Standard RWMS tablespaces are created using the `create_wms_tablespaces.sql` script located in `STAGING_DIR/rwms/installer/create_db`.

1. Modify `STAGING_DIR/rwms/installer/create_db/create_wms_tablespaces.sql`.
The table below shows the default initial sizes.

TABLESPACE_NAME	SIZE
ENCRYPTED_RETAIL_INDEX	12G
ENCRYPTED_RETAIL_DATA	10G
RETAIL_INDEX	10G
RETAIL_DATA	8G
LOB_DATA	2G
USERS	2G

2. Once this script has been modified, execute it in SQL*Plus as sys.
3. Review `create_wms_tablespaces.log` for errors and correct as needed.

Encrypted Tablespace Creation

If you do not have an Advanced Security Option license, create the `encrypted_retail_data` and `encrypted_retail_index` tablespaces as normal tablespaces.

1. Modify `STAGING_DIR/rwms/installer/create_db/create_encrypted_tablespaces_no_TDE.sql`.
2. Run the script using SQL*Plus as sys.
3. Review `Create_encrypted_retail_tablespaces_no_TDE.log` for errors and correct as needed.

With an Advanced Security license, tablespaces can be created in an encrypted format. The steps are:

Configure a Wallet:

1. Create a sqlnet.ora in \$TNS_ADMIN directory of the database server similar to the below entry:

```
ENCRYPTION_WALLET_LOCATION =
  (SOURCE = (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /u00/oracle/admin/ORACLE_SID/wallet)))
```

2. Create the wallet directory:

```
mkdir -p /u00/oracle/admin/<ORACLE_SID>/wallet
```

3. As a user with the 'alter system' privilege, create the wallet as follows:

Non-container databases:

- a. ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
'/u00/oracle/admin/dbName/wallet' IDENTIFIED BY "pwd#";
- b. KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "pwd#";
- c. KEY MANAGEMENT SET KEY IDENTIFIED BY "pwd#" WITH BACKUP;
- d. ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE
FROM KEYSTORE '/u00/oracle/admin/dbName/wallet' identified by pwd#;
 - a. Container databases:ADMINISTER KEY MANAGEMENT CREATE
KEYSTORE '/u00/oracle/admin/dbName/wallet' IDENTIFIED BY "pwd#";
 - b. ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE
FROM KEYSTORE '/u00/oracle/admin/dbName/wallet' identified by
"pwd#";
 - c. ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED
BY "pwd#" Container=ALL;
 - d. ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "pwd#"
WITH BACKUP USING 'TDE_ENCRYPTION' Container=all;

4. Confirm if the wallet is created and open (the TDE master encryption key has been created and inserted automatically):

```
select substr(wrl_type, 1, 10) wrl_type, substr(wrl_parameter, 1, 45) param, substr(status, 1, 10) status,
substr(wallet_type, 1, 15) w_type from v$encryption_wallet;
```

WRL_TYPE	PARAM	STATUS	W_TYPE
FILE	/u00/oracle/admin/ORACLE_SID/wallet	OPEN	AUTOLOGIN

An auto-open wallet is created. You are ready to create the encrypted tablespaces as shown in the following section.

Encryption at Tablespace Level

Once the wallet is configured, determine an encryption algorithm to use for the encrypted tablespace and then create them. The sample scripts use the default algorithm AES128:

1. Modify STAGING_DIR/rwms/installer/create_db/create_encrypted_tablespaces_TDE.sql.
2. Run the script using SQL*Plus as sys.
3. Review Create_encrypted_retail_tablespaces_TDE.log for errors and correct as needed.

Once the tablespaces have been created, the RMS schema installer can be run.

Note: After encryption at the tablespace level, it is absolutely crucial to backup the contents in the wallet directory; otherwise, if they are lost you will not be able to access the tablespaces.

Appendix: Sample Oracle Net Files for the Server

This appendix provides samples of Oracle Net files for the server.

listener.ora

Below is a sample listener.ora file.

\$SID represents the name of the Oracle instance that contains the RWMS schema.

Note: This example illustrates the listener configuration required for external procedures. It does not include environment specific settings that may be needed. Consult Oracle Net Services guides for additional information.

```
#####
# File: listener.ora
# Desc: Oracle Net8 listener file.
# Notes: Modify <hostname>
#####

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (PROTOCOL_STACK =
        (PRESENTATION = TTC)
        (SESSION = NS))
      (ADDRESS =
        (PROTOCOL = tcp)
        (HOST = <hostname>)
        (PORT = 1521))
      (ADDRESS =
        (PROTOCOL = IPC)
        (KEY = extproc_key))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (PROGRAM = extproc)
      (SID_NAME = extproc_agent)
      (ENVS='EXTPROC_DLLS=ANY')
    )
    (SID_DESC =
      (SID_NAME = <$SID>)
      (ORACLE_HOME = <$ORACLE_HOME>)
    )
  )
)
```

tnsnames.ora

Below is a sample tnsnames.ora file.

```
#####
# File: tnsnames.ora
# Desc: Net Services configuration file.
# Note: Change these values: <service_name>, <oracle_sid>, <hostname>,
#       <global_name>
#####
<service_name> =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = <hostname>)(Port = 1521)))
    (CONNECT_DATA = (SERVICE_NAME= <oracle_sservice_name>)))

<service_name>.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = <hostname>)(Port = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = <service_name>)))
```

Example:

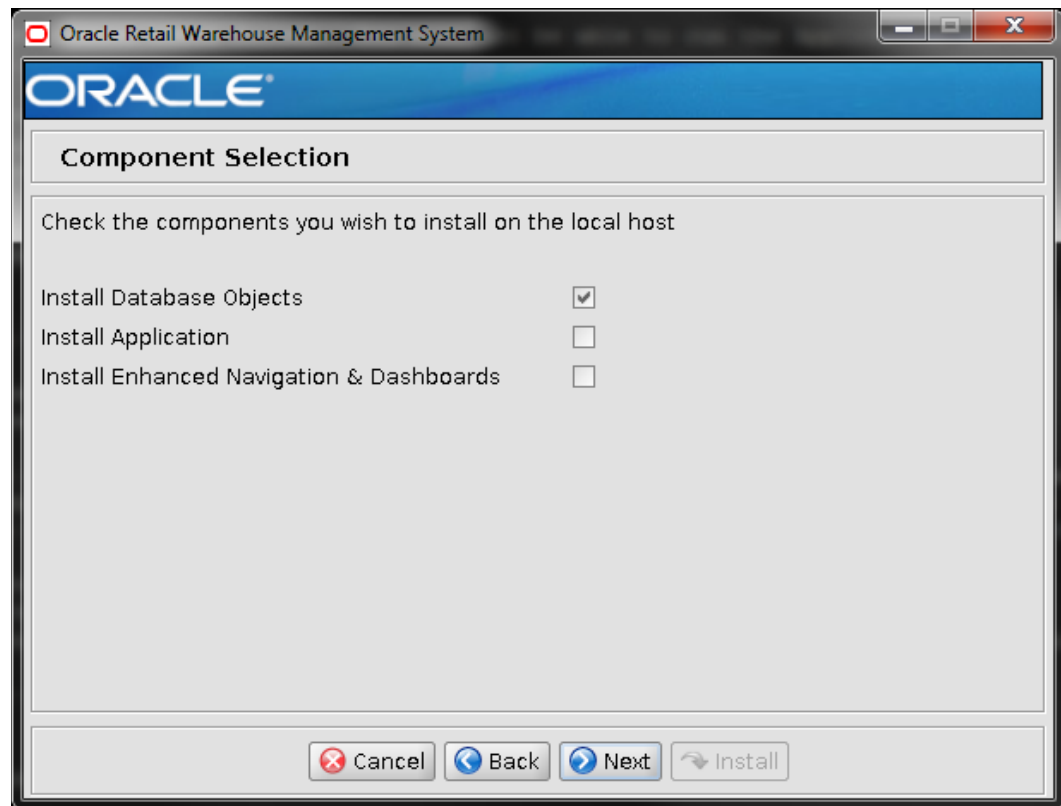
```
prod_dbl =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = server_01)(Port = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = prod_dbl)))

prod_dbl.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = server_01)(Port = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = prod_dbl) ()))
```

Appendix: RWMS Database Schema Installation Screens

You need the following details about your environment for the installer to successfully install the RWMS database schema. Depending on the options you select, you may not see some screens or fields.

Screen: Component Selection



Field Title	Component Selection.
Field Description	Select the RWMS component(s) you would like to install. Multiple components may be selected. You will not be able to install a component if the preinstall check for that component has failed.

Screen: Host Details

Oracle Retail Warehouse Management System

ORACLE

Host Details

Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname

Cancel Back Next Install

Field Title	Hostname
Field Description	Provide the hostname of the Oracle Database Server.
Examples	Dbhostname (db-host.us.oracle.com)

Screen: Oracle Database Details

Oracle Retail Warehouse Management System

ORACLE

Oracle Database Details

Please provide information on the database for this RWMS installation.

JDBC URL

Oracle SID

Cancel Back Next Install

Field Title	JDBC URL
Field Description	URL used by the installer to access the RWMS database schema. See Appendix: URL Reference for expected syntax.
Examples	<p>Standard Thin Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydb</code></p> <p>Standard Pluggable DB Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydbserviceName</code></p> <p>Standard OCI Connection: <code>jdbc:oracle:oci:@mydb</code></p> <p>RAC connection: <code>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname2)(PORT=1521))(LOAD_BALANCE=yes))(CONNECT_DATA=(SERVICE_NAME=mydb)))</code></p>

Field Title	Oracle SID
Field Description	Oracle system identifier for the database where the RWMS installation will run.
Examples	mydb

Screen: RWMS Database Schema Details

Oracle Retail Warehouse Management System

ORACLE

RWMS Database Schema Details

Please provide information on a pre-existing database user for this RWMS installation.

RWMS schema

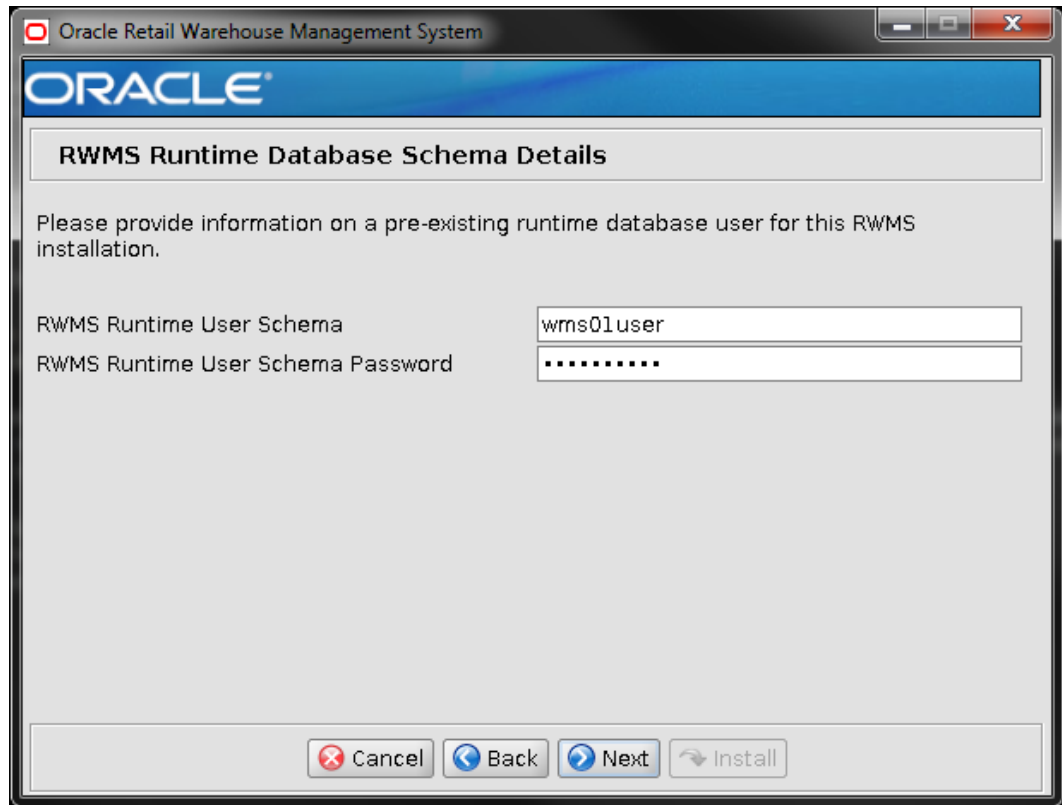
RWMS schema password

Cancel Back Next Install

Field Title	RWMS schema
Field Description	Provide the RWMS database owner here. The installer logs into the database as this user to create the RWMS schema. This user must already exist in the database when the RWMS database schema installation is run.
Examples	wms01

Field Title	RWMS schema password
Field Description	Database password for the RWMS schema owner.

Screen: RWMS Runtime Database Schema Details



Field Title	RWMS Runtime User Schema
Field Description	Provide the RWMS runtime database user here. This user must already exist in the database when the RWMS database schema installation is run. This will be used for running the RWMS application.
Examples	wms01user

Field Title	RWMS Runtime User Schema Password
Field Description	Database password for the RWMS Runtime user.

Screen: RWMS ADF Database Schema Details

Oracle Retail Warehouse Management System

ORACLE

RWMS ADF Database Schema Details

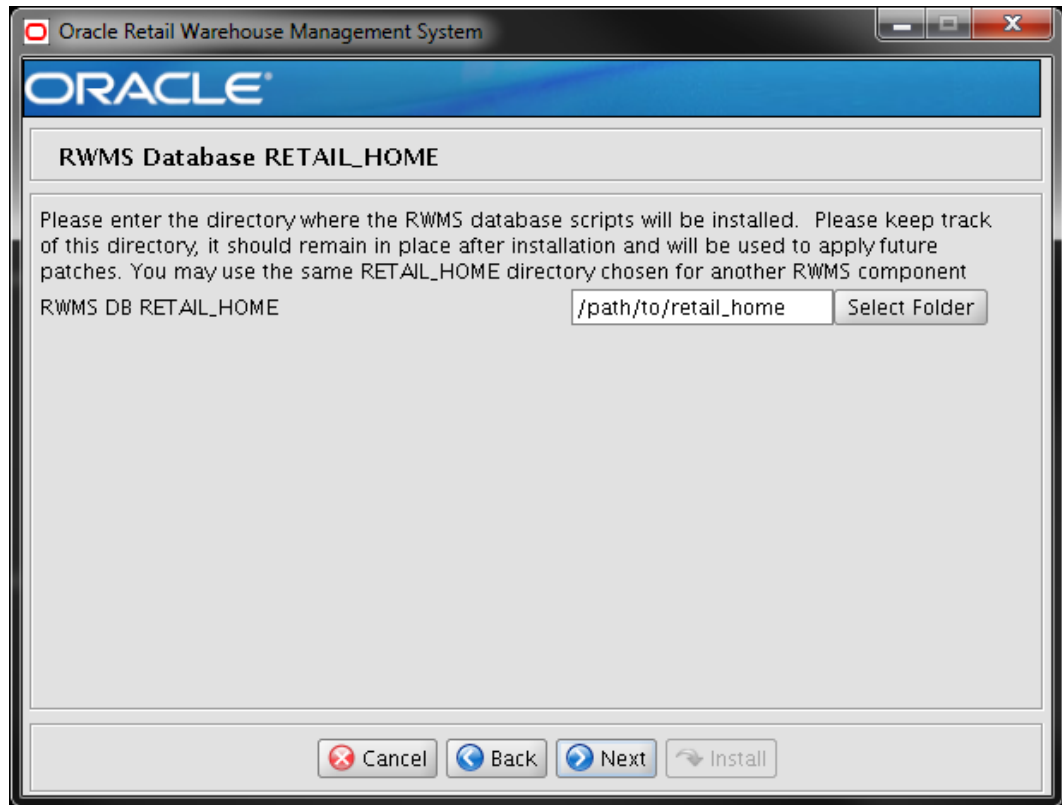
Please provide information on a pre-existing RWMS Enhanced Navigation & Dashboards ADF database user (RWMS_ADF_USER) for this RWMS installation.

RWMS ADF schema password

Cancel Back Next Install

Field Title	RWMS ADF schema password
Field Description	Database password for the RWMS ADF schema (RWMS_ADF_USER). This user must already exist in the database when the RWMS database schema installation is run. This is used for secure integration between ADF and Oracle Forms.

Screen: RWMS Database RETAIL_HOME



Field Title	RWMS DB RETAIL_HOME
Field Description	The location where the RWMS database scripts will be installed along with the ORPATCH utility.
Example	/path/to/retail_home
Note	If you have selected an existing RETAIL_HOME, and it has been configured to run other components than the ones you have selected for this installation, those components will also be installed regardless of what you selected on the Component Selection screen.

Screen: Oracle Wallet

Oracle Wallet

An Oracle Wallet is an encrypted container used to store and retrieve sensitive information, such as user credentials. Wallets are created if they don't already exist and configured to contain passwords used by RWMS. Every Wallet is itself protected by a password, and the field for this Wallet password must be filled out to move on to the next screen.

Note: If a wallet already exists for a RETAIL_HOME you have selected, this password must match the password for the existing wallet. Make sure this password is kept as it will be needed for future upgrades

The password must have a minimum length of eight characters and contain alphabetic characters combined with numbers or special characters.

Oracle Wallet password

Please re-enter password

Cancel Back Next Install

Field Title	Oracle Wallet password
Field Description	This is the password for the wallet that will store the credentials used during the RWMS installation. If you have selected an existing RETAIL_HOME in the previous screens, you will need to enter the password that was used for the wallet in that RETAIL_HOME.
Note	Make sure this password is kept as it will be needed for future upgrades.

Appendix: RWMS Application (forms) Installation Screens

Screen: Component Selection

Oracle Retail Warehouse Management System

ORACLE

Component Selection

Check the components you wish to install on the local host

Install Database Objects

Install Application

Install Enhanced Navigation & Dashboards

Cancel Back Next Install

Field Title	Component Selection.
Field Description	Select the RWMS component(s) you would like to install. Multiple components may be selected. You will not be able to install a component if the preinstall check for that component has failed.

Screen: Host Details

Oracle Retail Warehouse Management System

ORACLE

Host Details

Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname

Field Title	Hostname
Field Description	Provide the hostname of where the application needs to be installed.
Examples	apphostname (app-host.us.oracle.com)

Screen: Oracle Database Details

Oracle Retail Warehouse Management System

ORACLE

Oracle Database Details

Please provide information on the database for this RWMS installation.

JDBC URL

Oracle SID

Cancel Back Next Install

Field Title	JDBC URL
Field Description	URL used by the installer and application to access the RWMS database schema. See Appendix: URL Reference for expected syntax.
Examples	<p>Standard Thin Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydb</code></p> <p>Standard Pluggable DB Connection: <code>jdbc:oracle:thin:@dbhostname:1521/mydbserviceName</code></p> <p>Standard OCI Connection: <code>jdbc:oracle:thin:@mydb</code></p> <p>RAC connection: <code>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=dbhostname2)(PORT=1521))(LOAD_BALANCE=yes))(CONNECT_DATA=(SERVICE_NAME=mydb)))</code></p>

Field Title	Oracle SID
Field Description	Oracle system identifier for the database that the RWMS installation will connect to. You could enter in either the Oracle SID or the Database service name if using a pluggable database.
Examples	mydb

Screen: RWMS Database Schema Details

Oracle Retail Warehouse Management System

ORACLE

RWMS Database Schema Details

Please provide information on a pre-existing database user for this RWMS installation.

RWMS schema

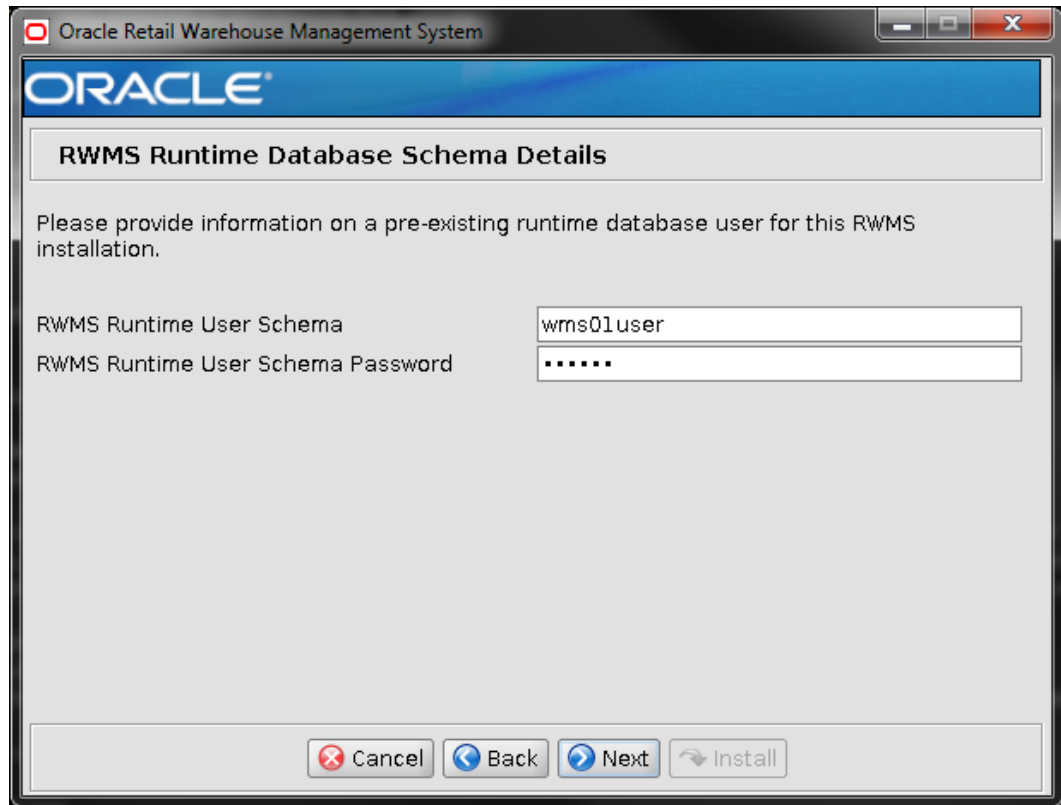
RWMS schema password

Cancel Back Next Install

Field Title	RWMS schema
Field Description	Provide the RWMS database owner here. The installer uses this schema to compile forms.
Examples	wms01

Field Title	RWMS schema password
Field Description	Database password for the RWMS schema owner.

Screen: RWMS Runtime Database Schema Details



Field Title	RWMS Runtime User Schema
Field Description	Provide the RWMS runtime database user here.. This user must already exist in the database when the RWMS database schema installation is run. This will be used for running the RWMS application.
Examples	wms01user

Field Title	RWMS Runtime User Schema Password
Field Description	Database password for the RWMS Runtime user.

Screen: Installation Name

Oracle Retail Warehouse Management System

ORACLE

Installation Name

Please enter the Installation Name. The Installation Name is a unique name that will be used to identify this installation of RWMS in Oracle Forms and will be included in the Forms Application URLs. This name is also used to identify the RWMS Enhanced Navigation & Dashboards application in Weblogic.

Application Installation Name

Cancel Back Next Install

Field Title	Application Installation Name
Field Description	The value used to configure and identify a specific forms installation. The Forms URLs for this installation will contain this value.
Example	rwmsinst
Note	The value entered in this field must be identical for the Application (Forms) and Enhanced Navigation & Dashboards installations.

Screen: Forms WebLogic Administrative Details

Oracle Retail Warehouse Management System

ORACLE

Forms Weblogic Administrative Details

Enter the administrative user and password for the Forms Weblogic Server.

Forms Weblogic Admin port: 7001

Forms Weblogic Admin User: weblogic

Forms Weblogic Admin Password:

Please re-enter password:

Enable SSL for RWMS Forms? Yes No

Note: enabling SSL requires that security certificates have been configured and installed for this WebLogic domain. The AdminServer and all managed servers must then be configured to use SSL.

Cancel Back Next Install

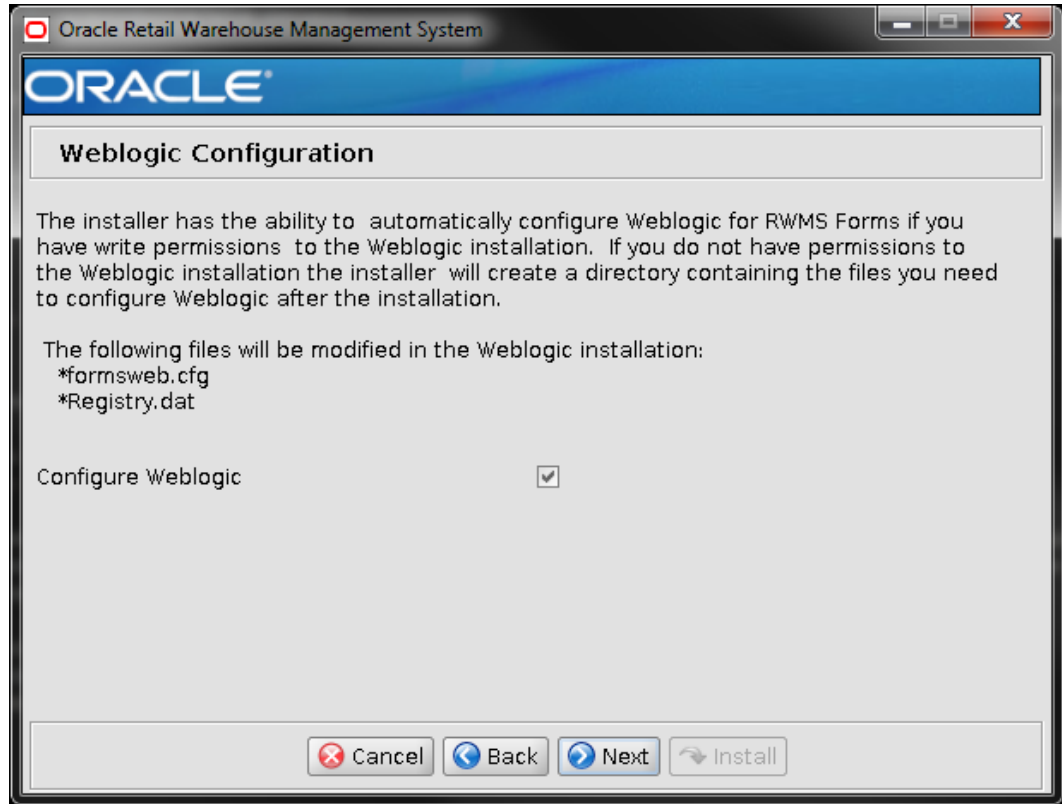
Field Title	Forms WebLogic Admin Port
Field Description	Listen port for the Forms WebLogic Admin server.
Example	7001

Field Title	Forms WebLogic Admin User
Field Description	Username of the admin user for the WebLogic instance to which the Oracle Forms application is already deployed.
Example	weblogic

Field Title	Forms WebLogic Admin Password
Field Description	Password for the WebLogic admin user. You chose this password when you created the WebLogic instance.

Field Title	Enable SSL for RWMS Forms?
Field Description	Choose Yes to install RWMS Forms using a WebLogic environment configured to use SSL. In this case, SSL must be configured and the ports must be enabled for the AdminServer and Oracle Forms managed servers. Choose No to install using a WebLogic environment configured without SSL. In this case the non-SSL ports must be enabled for the AdminServer and for the Oracle Forms managed servers.

Screen: WebLogic Configuration



Field Title	Configure WebLogic
Field Description	Make the necessary configurations to the WebLogic server to be able to run RWMS forms. If you choose no, the configured WebLogic files will be generated by the installer, but should be applied to WebLogic manually.

Screen: RWMS Application RETAIL_HOME

Oracle Retail Warehouse Management System

ORACLE

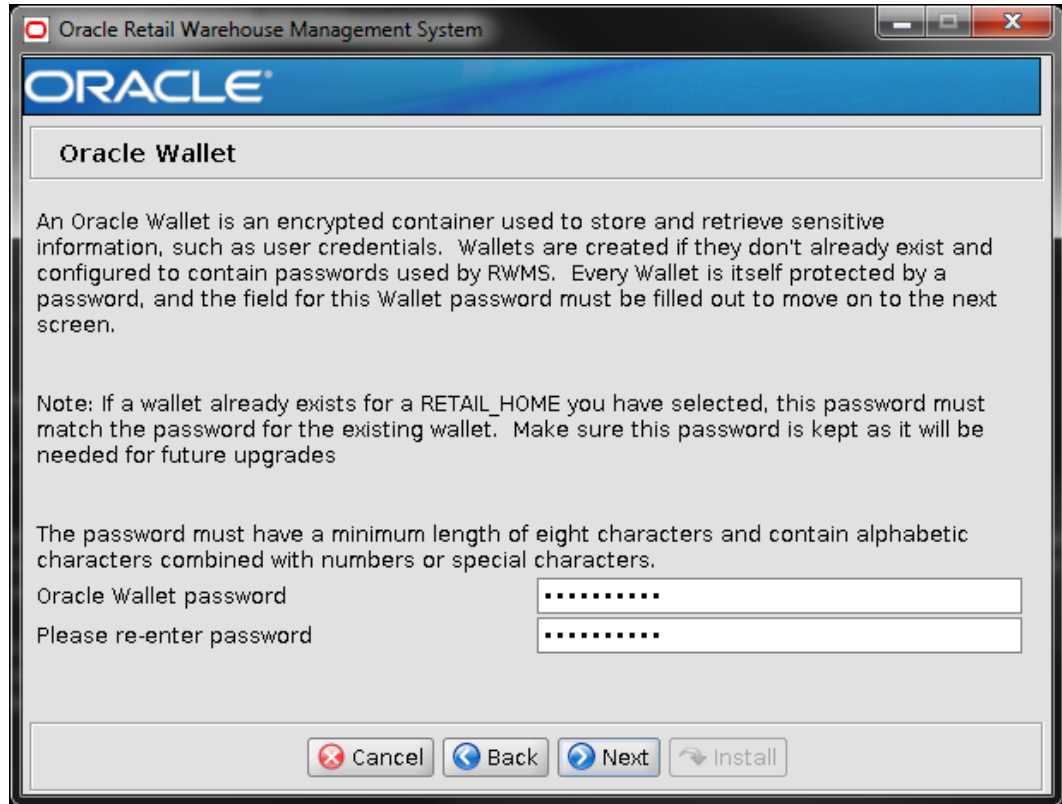
RWMS Application RETAIL_HOME

Please enter the directory where RWMS Application forms will be installed. Typically the RWMS forms installation directory is located outside of the Weblogic installation. Please keep track of this directory, it should remain in place after installation and will be used during Enhanced Navigation & Dashboards installation. It will also be used to apply future patches. You may use the same RETAIL_HOME directory chosen for another RWMS component

RWMS Application RETAIL_HOME

Field Title	RWMS Application RETAIL_HOME
Field Description	The location where the RWMS Application (forms and reports) will be installed along with the ORPATCH utility. The RWMS Forms will be installed in a subdirectory of this directory, named base.
Example	/path/to/retail_home
Note	If you have selected an existing RETAIL_HOME, and it has been configured to run other components than the ones you have selected for this installation, those components will also be installed regardless of what you selected on the Component Selection screen.

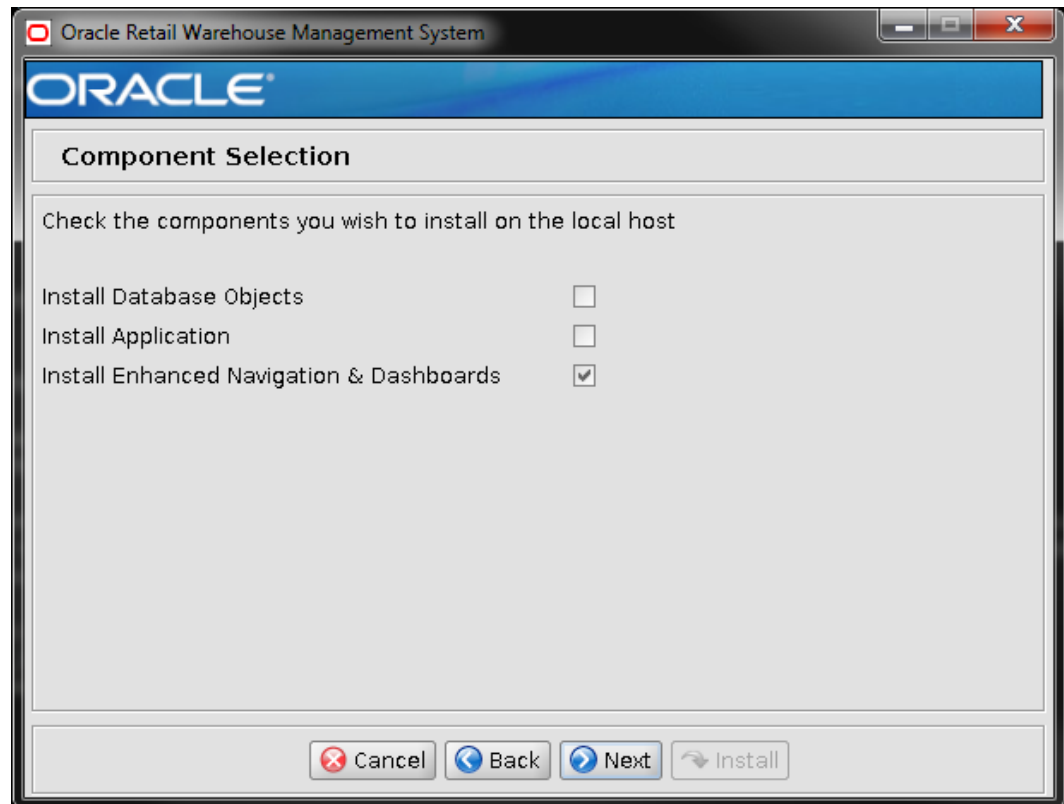
Screen: Oracle Wallet



Field Title	Oracle Wallet password
Field Description	This is the password for the wallet that will store the credentials used during the RWMS installation. If you have selected an existing RETAIL_HOME in the previous screens, you will need to enter the password that was used for the wallet in that RETAIL_HOME.
Note	Make sure this password is kept as it will be needed for future upgrades.

Appendix: RWMS Enhanced Navigation and Dashboards Installation Screens

Screen: Component Selection



Field Title	Component Selection.
Field Description	Select the RWMS component(s) you would like to install. Multiple components may be selected. You will not be able to install a component if the preinstall check for that component has failed.

Screen: Host Details

Oracle Retail Warehouse Management System

ORACLE

Host Details

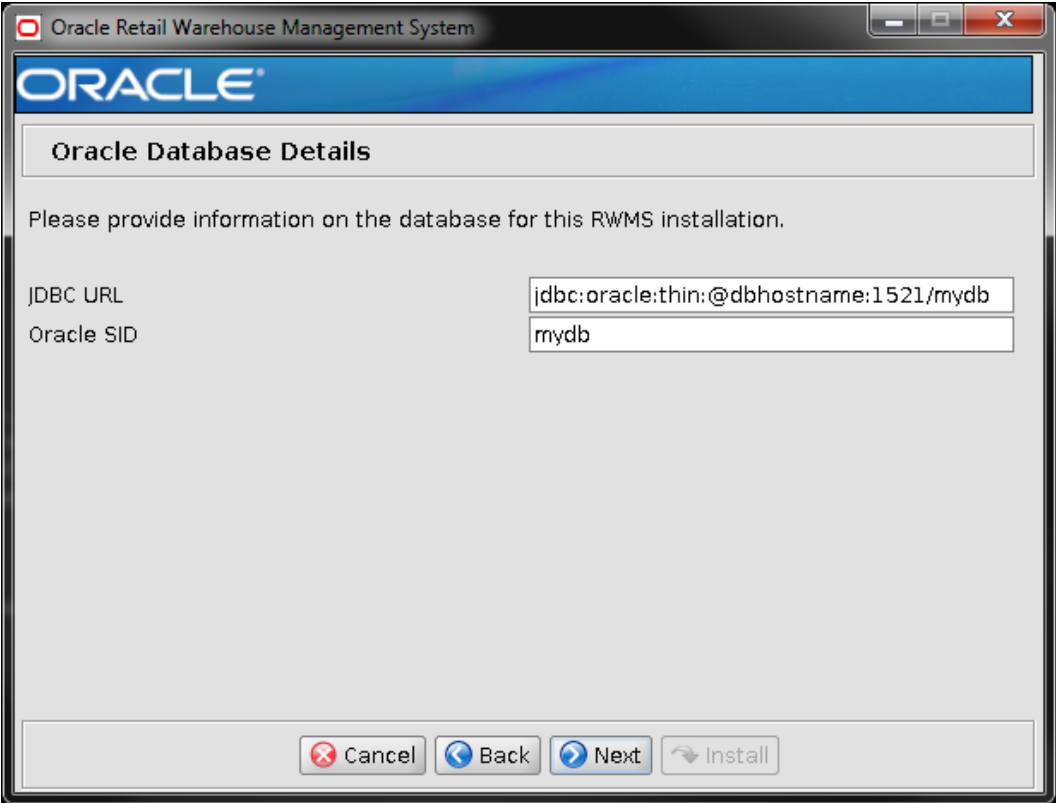
Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname

Cancel Back Next Install

Field Title	Hostname
Field Description	Provide the hostname of where the application needs to be installed.
Examples	apphostname (app-host.us.oracle.com)

Screen: Oracle Database Details



Field Title	JDBC URL
Field Description	URL used by the application to access the RWMS database schema. See Appendix: URL Reference for expected syntax.
Examples	<p>Standard Thin Connection: jdbc:oracle:thin:@dbhostname:1521/mydb</p> <p>Standard Pluggable DB Connection: jdbc:oracle:thin:@dbhostname:1521/mydbserviceName</p> <p>Standard OCI Connection: jdbc:oracle:thin:@mydb</p> <p>RAC connection: jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP)(HOST = dbhostname1)(PORT = 1521))(ADDRESS = (PROTOCOL = TCP)(HOST = dbhostname2)(PORT = 1521))(LOAD_BALANCE = yes))(CONNECT_DATA =(SERVICE_NAME =mydb)))</p>
Note	Do not use an oci url for an Enhanced Navigation & Dashboards installation. The thin url should be used instead, as shown in the screenshot above. If you are using the same installer STAGING_DIR that you used previously for an RWMS Application install, it may default to the previously used oci jdbc url and you will have to change it.

Field Title	Oracle SID
Field Description	Oracle system identifier for the database that the RWMS installation will connect to.
Examples	mydb

Screen: RWMS Runtime Database Schema Details

Oracle Retail Warehouse Management System

ORACLE

RWMS Runtime Database Schema Details

Please provide information on a pre-existing runtime database user for this RWMS installation.

RWMS Runtime User Schema

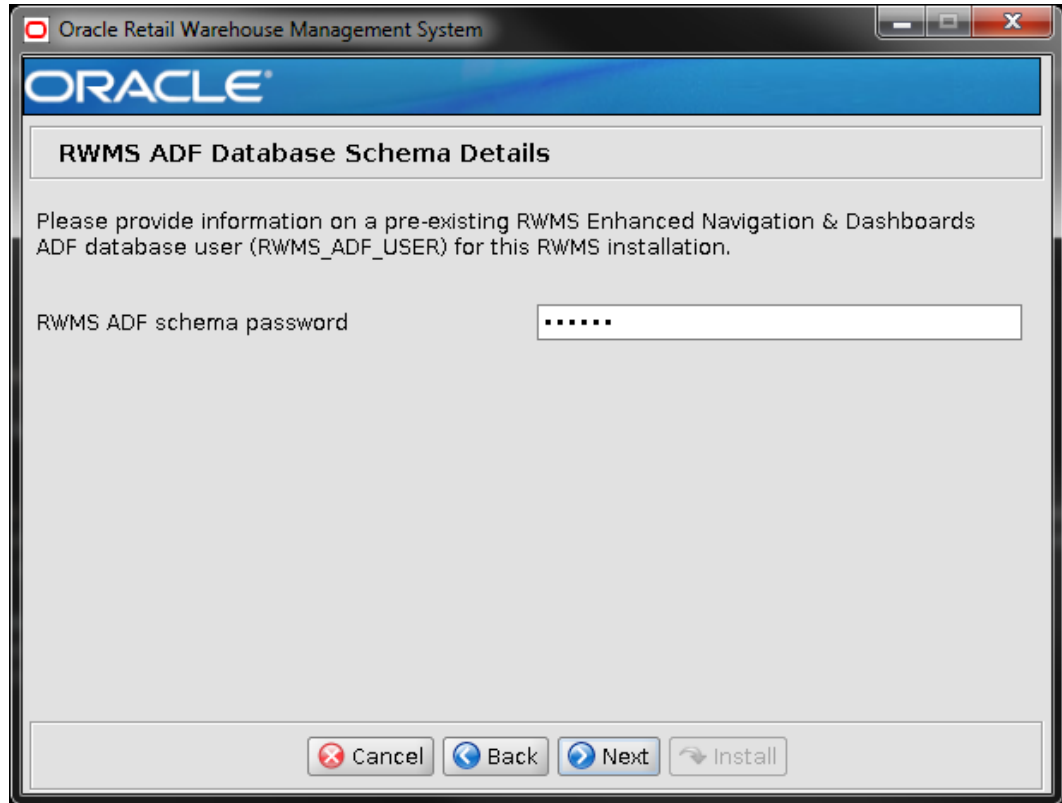
RWMS Runtime User Schema Password

Cancel Back Next Install

Field Title	RWMS Runtime User Schema
Field Description	Provide the RWMS runtime database user here.. This user must already exist in the database when the RWMS database schema installation is run. This will be used for running the RWMS application.
Examples	wrms01user

Field Title	RWMS Runtime User Schema Password
Field Description	Database password for the RWMS Runtime user.

Screen: RWMS ADF Database Schema Details



Field Title	RWMS ADF schema password
Field Description	Database password for the RWMS ADF schema (RWMS_ADF_USER). This is used for secure integration between ADF and Oracle Forms.

Screen: Installation Name

Oracle Retail Warehouse Management System

ORACLE

Installation Name

Please enter the Installation Name. The Installation Name is a unique name that will be used to identify this installation of RWMS in Oracle Forms and will be included in the Forms Application URLs. This name is also used to identify the RWMS Enhanced Navigation & Dashboards application in Weblogic.

Application Installation Name

Cancel Back Next Install

Field Title	Application Installation Name
Field Description	The value used as the WebLogic deployment name for the Enhanced Navigation & Dashboards application and to configure this application to connect to the RWMS Forms installation.
Example	rwmsinst
Note	The value entered in this field must be identical for the Application (Forms) and Enhanced Navigation & Dashboards installations.

Screen: Enhanced Navigation & Dashboards JDBC Security details



Field Title	Enable Secure JDBC connection
Field Description	Choose "Yes" to create secured data sources in WebLogic, otherwise choose "No". A secure data base connection must already be set up if you want to create a secure data source.
Note	Refer to Security Guide for information on how to secure the database connection.

Screen: Enhanced Navigation & Dashboards Secure Data Source Details

Field Title	Identity Keystore
Field Description	Path to the identity keystore used for a secure connection to the RWMS datasource.
Example	/path/to/.keystore
Note	Refer to the Security Guide for more information.

Field Title	Identity Keystore Type
Field Description	Type of the identity keystore used.
Example	JKS

Field Title	Identity Keystore Password
Field Description	Password used to access the identity keystore defined above.

Field Title	Identity Truststore
Field Description	Path to the identity truststore used for a secure connection to the RWMS datasource.
Example	/path/to/.keystore

Field Title	Identity Truststore Type
Field Description	Type of the identity truststore used.
Example	JKS

Field Title	Identity Truststore Password
Field Description	Password used to access the identity truststore defined above.

Screen: Enhanced Navigation & Dashboards Secure Data Source Details

Field Title	Identity Keystore
Field Description	Path to the identity keystore used for a secure connection to the RWMS ADF datasource.
Example	/path/to/.keystore
Note	Refer to the Security Guide for more information

Field Title	Identity Keystore Type
Field Description	Type of the identity keystore used.
Example	JKS

Field Title	Identity Keystore Password
Field Description	Password used to access the identity keystore defined above.

Field Title	Identity Truststore
Field Description	Path to the identity truststore used for a secure connection to the RWMS ADF datasource.
Example	/path/to/.keystore

Field Title	Identity Truststore Type
Field Description	Type of the identity truststore used.
Example	JKS

Field Title	Identity Truststore Password
Field Description	Password used to access the identity truststore defined above.

Screen: Enhanced Navigation & Dashboards Weblogic Administrative Details

Enhanced Navigation & Dashboards Weblogic Administrative Details

Enter the administrative user and password for the Weblogic Server to which the Enhanced Navigation & Dashboards application will be deployed.

WebLogic Admin Port: 18001

Weblogic Admin User: weblogic

Weblogic Admin Password:

Please re-enter password:

Enter the RWMS Enhanced Navigation & Dashboards Weblogic managed server or cluster.

RWMS Server/Cluster: rwms-server

Enable SSL? Yes No

Note: enabling SSL requires that security certificates have been configured and installed for this WebLogic domain. The AdminServer and all managed servers must then be configured to use SSL.

Buttons: Cancel, Back, Next, Install

Field Title	WebLogic Admin Port
Field Description	Listen port for the Enhanced Navigation & Dashboards WebLogic Admin server.
Example	18001

Field Title	WebLogic Admin User
Field Description	Username of the admin user for the WebLogic instance to which the Enhanced Navigation & Dashboards application will be deployed.
Example	weblogic

Field Title	WebLogic Admin Password
Field Description	Password for the WebLogic admin user. You chose this password when you created the WebLogic instance.

Field Title	RWMS 14 Server/Cluster
Field Description	The Server or the cluster name where the Enhanced Navigation and Dashboards application is to be installed.
Examples	rwms-server

Field Title	Enable SSL?
Field Description	Choose Yes to install Enhanced Navigation & Dashboards to a WebLogic environment configured to use SSL. In this case, SSL must be configured and the ports must be enabled for the AdminServer and RWMS Server/Cluster. Choose No to install to a WebLogic environment configured without SSL. In this case the non-SSL ports must be enabled for the AdminServer and for the RWMS Server/Cluster.

Screen: RWMS Application RETAIL_HOME

Oracle Retail Warehouse Management System

ORACLE

RWMS Application RETAIL_HOME

Please enter the RETAIL_HOME directory where RWMS Application forms have already been installed.

RWMS Application RETAIL_HOME

Cancel Back Next Install

Field Title	RWMS Application RETAIL_HOME
Field Description	The location where the RWMS Application (forms and reports) will be installed along with the ORPATCH utility. The RWMS Forms will be installed in a subdirectory of this directory, named base.
Examples	/path/to/forms/retail_home

Screen: Enhanced Navigation & Dashboards RWMS Application details

The screenshot shows a configuration window titled "Oracle Retail Warehouse Management System" with a sub-header "Enhanced Navigation & Dashboards RWMS Application Details". The window contains the following fields and options:

- Forms Hostname, Port & Reports Url**: A section header.
- Forms connection protocol**: Radio buttons for http and https.
- Forms hostname**: Text input field containing "apphostname".
- Forms port**: Text input field containing "9001".
- Reports URL**: Text input field containing "https://bihostname:9001/xmlpserver/servlet/rc".

At the bottom of the window are four buttons: "Cancel", "Back", "Next", and "Install".

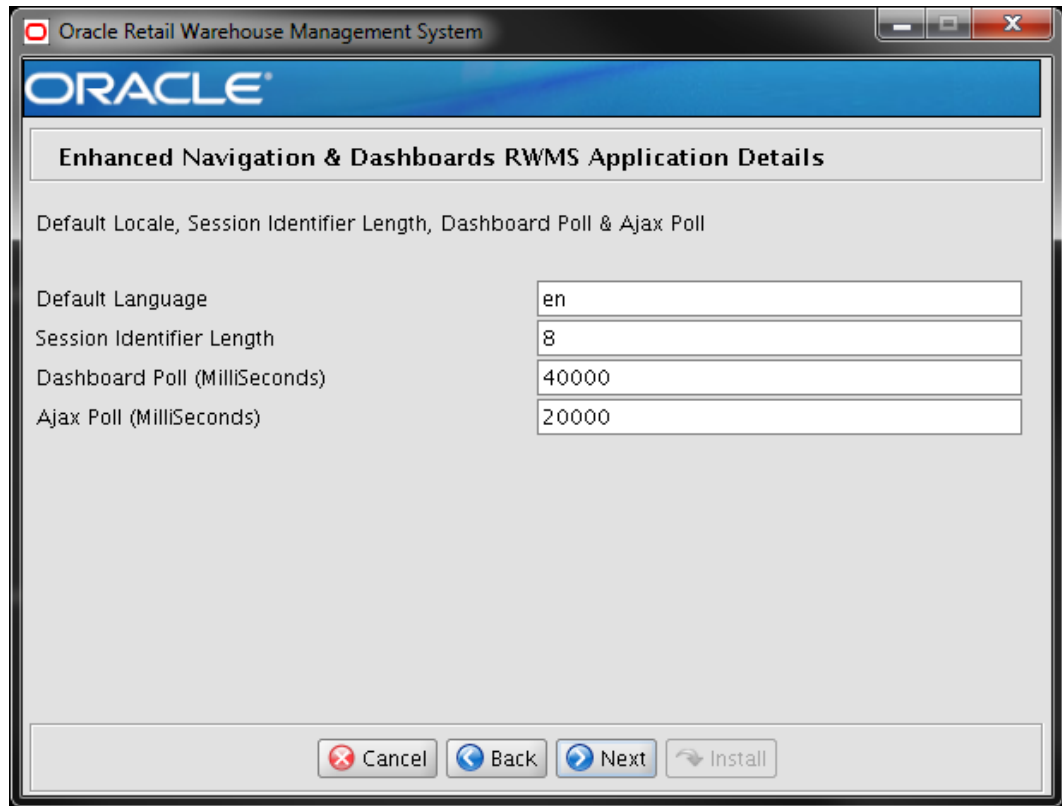
Field Title	Forms connection protocol
Field Description	Select the correct protocol value depending on whether or not RWMS forms was installed using a WebLogic environment configured to use SSL. This should match the selection chosen for the field "Enable SSL for RWMS Forms?" used during the RWMS Forms Application installation. If "Yes" was chosen, select "https". If "No" was chosen, select "http".

Field Title	Forms hostname
Field Description	Provide the hostname where the RWMS Forms application is installed.
Examples	apphostname (app-host.us.oracle.com)

Field Title	Forms Port
Field Description	Listen port for the WebLogic managed server where Oracle Forms is deployed.
Examples	9001

Field Title	Reports URL
Field Description	<p>Provide the URL to access RWMS Reports. This should be setup as follows: https://<bihostname>:<port>/xmlpserver/servlet/report?f=/Guest/RWMS/RWMS+REPOSITORY</p> <p>NOTE: This value is going to be used in the .env file to provide a value for the RWMS_REPORTS_URL variable. This field can be left blank but would need to be updated later as a post install step.</p>
Examples	https://bihostname:9001/xmlpserver/servlet/report?f=/Guest/RWMS/RWMS+REPOSITORY

Screen: Enhanced Navigation & Dashboards RWMS Application Details



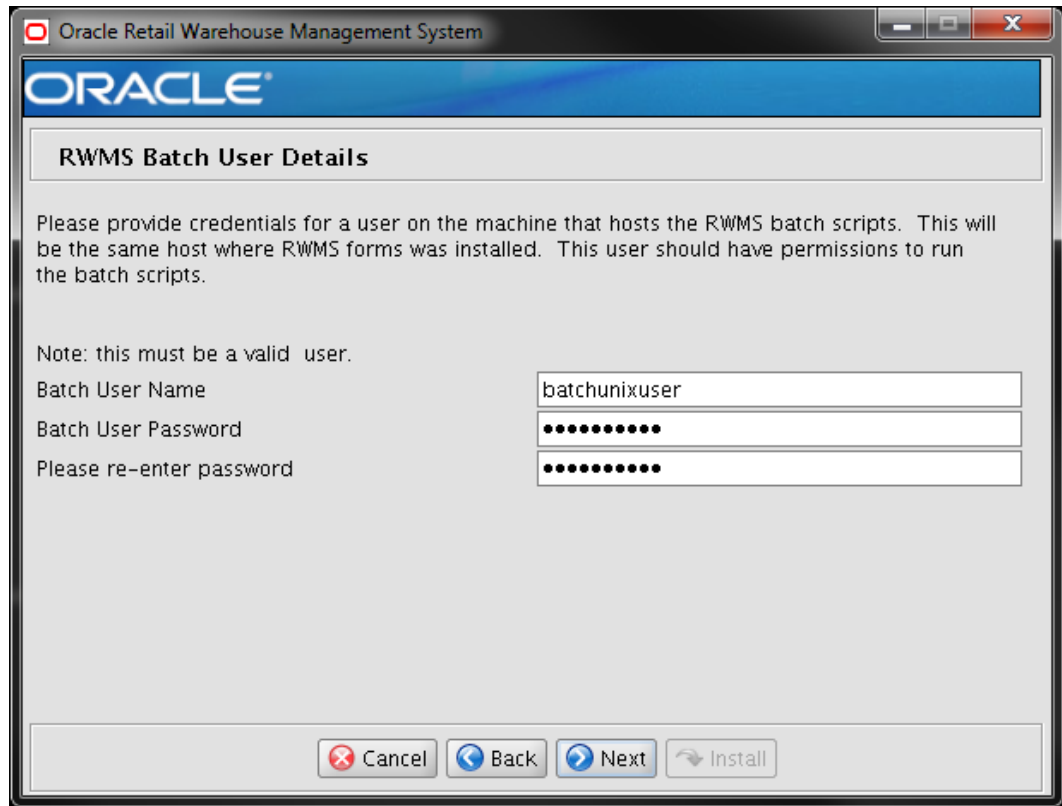
Field Title	Default Language
Field Description	Language code used in RWMS to translate the login page.
Examples	en

Field Title	Session Identifier Length
Field Description	Specifies the length of the alpha numeric session key generated by the system during login. The recommended value is 8.
Examples	8

Field Title	Dashboard Poll (Milliseconds)
Field Description	Dashboard database polling frequency, in milli-seconds.
Examples	40000

Field Title	Ajax Poll (Milliseconds)
Field Description	Ajax polling frequency for browser calls to the shared dashboard server model, in milli-seconds.
Examples	20000

Screen: RWMS Batch User Details



Field Title	Batch User Name
Field Description	The username used to connect to the machine that hosts the RWMS batch scripts and run them. The user should have the necessary permissions to write to the <retail_home>/log directory and execute shell scripts in the <retail_home>/forms/bin directory. Those files are created and will have permissions set according to the umask of the system user that is running the installer. A post install sep may be needed to update the permissions on those files in order for the batch user entered in this screen to be able to write and execute scripts.
Examples	batchunixuser

Field Title	Batch User Password
Field Description	The password for the user entered for Batch User Name

Appendix: RWMS Analyze Tool

It may be desirable to see a list of the files that will be updated by a patch, particularly if files in the environment have been customized. The installer has an ‘analyze’ mode that will evaluate all files in the patch against the environment and report on the files that will be updated based on the patch. See the section “Analyzing the Impact of a Patch” in the chapter “RWMS Patching Procedures” for more details.

Run the RWMS Analyze Tool

1. Log onto the server as a user with access to the RETAIL_HOME for the installation you want to analyze.
2. Change directories to STAGING_DIR/rwms/installer. STAGING_DIR is the location where you extracted the 15.0.x installer.
3. Set and export the following environment variables.

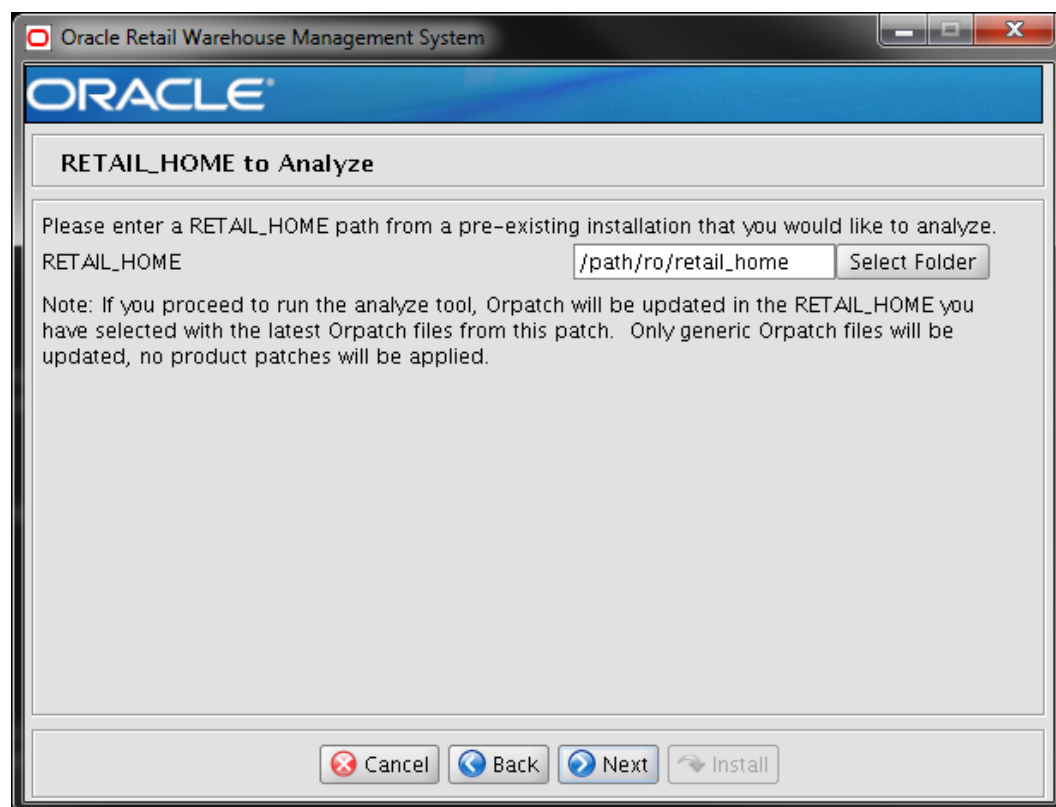
Variable	Description	Example
JAVA_HOME	Location of a Java 1.7+ 64Bit JDK.	JAVA_HOME= /u00/webadmin/java/jdk1.7.0 export JAVA_HOME
DISPLAY	Address and port of X server on desktop system of user running install. Optional when running the Analyze tool	DISPLAY=<IP address>:0.0 export DISPLAY

4. If you are going to run the installer in GUI mode using an X server, you need to have the XTEST extension enabled. This setting is not always enabled by default in your X server. See [Appendix: Common Installation Errors](#) for more details.
5. Run the analyze.sh script to start the analyze tool.

Note: Below are the usage details for analyze.sh. The typical usage for GUI mode is no arguments.

```
./analyze.sh [text | silent]
```

Screen: RETAIL_HOME to Analyze



Field Title	RETAIL_HOME
Field Description	The pre-existing location where RWMS (database and/or application) was installed along with the ORPATCH utility. This location should contain directories with your installed files as well as the "orpatch" directory.
Example	/path/to/retail_home
Note	The Orpatch files in this RETAIL_HOME may need to be updated in order to be able to run the analysis. The Analyze tool will take care of this automatically.

6. After clicking "install", the Analyze tool will generate a report of the files that will be patched if you apply this patch to the selected RETAIL_HOME. A high level report can be found in the log file: STAGING_DIR/rwms/installer/logs/rwms-analyze.<timestamp>.log.
7. The detailed list of patch files can be found in RETAIL_HOME/orpatch/logs/detail_logs/analyze/details/

Appendix: Installer Silent Mode

In addition to the GUI and text interfaces of the installer, there is a silent mode that can be run. This mode is useful if you wish to run a repeat installation without retyping the settings you provided in the previous installation. It is also useful if you encounter errors in the middle of an installation and wish to continue.

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

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

1. Edit the `ant.install.properties` file and correct any invalid settings that may have caused the installer to fail in its previous run.
2. Look for duplicate properties in the `ant.install.properties` file. Some properties are set on multiple pages to ensure default values when a page is only displayed under certain conditions. For example, if there are two instances of `input.property.name`, remove all but the last one.
3. Run the installer again with the **silent** argument.

Example: `install.sh silent`

Appendix: Common Installation Errors

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

Database Installer Hangs on Startup

Symptom

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

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

Solution

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

Warning: Could not create system preferences directory

Symptom

The following text appears in the installer Errors tab:

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

Solution

This is related to Java bug 4838770. The `/etc/.java/.systemPrefs` directory may not have been created on your system. See <http://bugs.sun.com> for details.

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

WLS_FORMS server failed to restart

Symptom

The Application installer fails saying the Forms server couldn't restart with the bellow error.

Error was:

```
[exec] This Exception occurred at Thu Nov 14 04:20:39 EST 2013.  
[exec] javax.naming.CommunicationException [Root exception is  
java.net.ConnectException: t3s://msp52420:7002: Destination unreachable; nested  
exception is:  
[exec] javax.net.ssl.SSLKeyException: [Security:090504]Certificate chain received  
from msp52420 - 10.141.53.240 failed hostname verification check. Certificate contained  
msp52420.us.oracle.com but check expected msp52420; No available router to  
destination]
```

Solution

Provide complete hostname in the "Host Details" installer screen (msp54720.us.oracle.com instead of msp54720) , and the install will go through successfully.

Warning: Couldn't find X Input Context

Symptom

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

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

Solution

This message is harmless and can be ignored.

FRM-30085: Unable to adjust form for output

Symptom

When compiling forms, compilation fails with the error FRM-30085: Unable to adjust form for output. The forms log has either nothing showing in the output as an error, or it fails with cannot load library unit PM_RETAIL_API_SQL (referenced by INTERNAL_VARIABLES).

PM_RETAIL_API could be a variety of different packages.

Solution

Run the below query as sys and compile the output until no rows are returned

```
SQL> set heading off SQL> select 'alter '||do.object_type||' ' || owner ||
'. '||do.object_name||' compile;'
from dba_objects do,diana_version$ dv where dv.obj#=do.object_id and do.timestamp
<> to_char(dv.stime,'YYYY-MM-DD:HH24:MI:SS')
/
alter FUNCTION RWMS15MOCK.ASN_DISTRIBUTE_UPS compile;
alter FUNCTION RWMS15MOCK.CREATE_BLOCK_GROUP compile;

SQL> alter FUNCTION RWMS15MOCK.ASN_DISTRIBUTE_UPS compile;

Function altered.

SQL> alter FUNCTION RWMS15MOCK.CREATE_BLOCK_GROUP compile;

Function altered.

SQL> select 'alter '||do.object_type||' ' || owner || '.'||do.object_name||'
compile;' from dba_objects do,diana_version$ dv
where dv.obj#=do.object_id 4 and do.timestamp <> to_char(dv.stime,'YYYY-MM-
DD:HH24:MI:SS')
/
no rows selected

SQL>
```

When no rows are returned, rerun the installer.

Unresponsive Drop-Downs

Symptom

In GUI mode, when you click on the drop-down list, the list does not appear, and the following message appears in the console window:

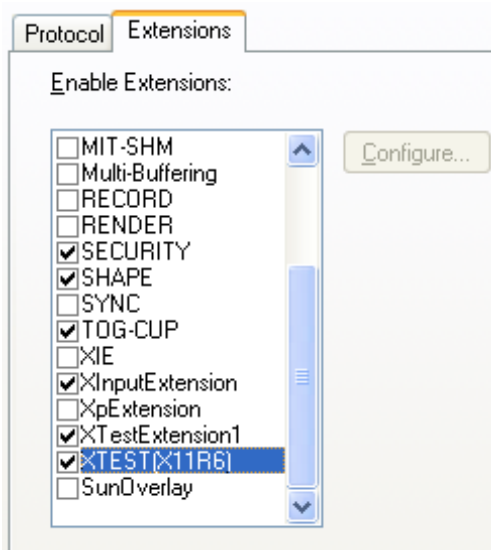
```
XTEST extension not installed on this X server: Error 0
```

Solution

To run the installer in GUI mode, you must have the XTEST extension enabled in your X server.

Enabling XTEST in Exceed:

1. Open Xconfig to edit Exceed configuration.
2. Go to the X Server Protocol settings.
3. Click the Extensions tab.
4. Make sure that the XTEST extension is selected:



5. Restart the X Server and run the installer again.

Could not exec robot child process: Permission denied

Symptom

When opening a drop-down list in GUI mode of the RWMS installer, the installer freezes up and displays the following message in the console:

```
Couldn't exec robot child process: Permission denied
```

Solution

As the owner of the database ORACLE_HOME (i.e. *oracle*), grant execute permissions to the `awt_robot*` files under `$ORACLE_HOME/jdk/jre/lib`. The database schema installer uses `$ORACLE_HOME/jdk` for its `JAVA_HOME`.

Example (using SUN Solaris):

```
chmod a+x $ORACLE_HOME/jdk/jre/lib/sparc/awt_robot
chmod a+x $ORACLE_HOME/jdk/jre/lib/sparcv9/awt_robot
```

ConcurrentModificationException in Installer GUI

Symptom

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

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

Solution

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

ORA-04031 (unable to allocate memory) error during database schema installation

Symptom

When running the database schema installer you get the following error one or more times:

```
[ora:sqlplus] alter package
[ora:sqlplus] *
[ora:sqlplus] ERROR at line 1:
[ora:sqlplus] ORA-04031: unable to allocate 92120 bytes of shared memory ("shared
[ora:sqlplus] pool", "unknown object", "PL/SQL MPCODE", "BAMIMA: Bam Buffer")
```

Solution

There was not enough available memory in the shared pool on the database at the time of compilation. There are several choices to get past this error:

- Log into the database and attempt to recompile invalid objects in the database schema. Subsequent attempts to compile the same object(s) can be successful.
- Have a DBA increase the shared pool size on the database and re-run the installer from scratch on a new schema user.

X Error of failed request: BadWindow (invalid Window parameter)

Symptom

When compiling forms during the application installation you receive this error one or more times:

```
X Error of failed request: BadWindow (invalid Window parameter)
Major opcode of failed request: 18 (X_ChangeProperty)
Resource id in failed request: 0x1800002
Serial number of failed request: 432
Current serial number in output stream: 437
```

Solution

This error occurs when there are too many requests made to the X server. If this error occurs manually recompile the form.

Example:

```
frmpcmp.sh userid=$SUP module_type=form module=FORM_OR_MENU
```

RIB Errors

At random times, the RIB will get certain errors such as GETNEXT(?,?,?,?,?) and/or ORA-21700 object does not exist or is marked for delete. This is very confusing because you may research and find that the object exists and is valid.

You must re-initialize the reference to reference an existing object. You do this by:

1. Bringing down the RIB OAS in question
2. Running /RIB_INSTALL_DIR>/InstallAndCompileAllRibOracleObjects.sql
3. Running another object validate script (ex: inv_obj_comp.sql) to make sure objects are valid (some may have deadlocked in the end of the previous step).
4. Bringing up the RIB OAS in question

Error Connecting to Database URL

Symptom

After entering database credentials in the installer screens and hitting next, a message pops up with an error like this:

```
Error connecting to database URL <url> as user <user>
details...
```

The message prevents you from moving on to the next screen to continue the installation.

Solution

This error occurs when the installer fails to validate the user credentials you have entered on the screen. Make sure that you have entered the credentials properly. If you receive a message similar to this:

```
Error connecting to database URL <url> as user <user>
java.lang.Exception: UnsatisfiedLinkError encountered when using the Oracle
driver.
```

Please check that the library path is set up properly or switch to the JDBC thin client.

It may mean that the installer is using the incorrect library path variables for the platform you are installing on. Open the file

```
<STAGING_DIR>/rwms/dbschema/common/preinstall.sh and toggle the variable
"use32bit" to "true" if it is set to "false" or vice versa. This setting is dependent on the
JRE that is being used.
```

Cannot Run Reports in RWMS Application on HP Host

Symptom

When RWMS application is installed on HP - The user will receive errors when attempting to execute reports from within the RWMS application.

Solution

This is a known issue for RWMS on HP operating systems – The jar files that drive the printing functionality do not work. This is not an issue for any of the other operating systems that RWMS is certified to run on.

Frequent Session Timeouts in RWMS Application on HP Host

Symptom

When RWMS application is installed on HP - The user will frequently get kicked out the application with a message indicating that the users session has timed out and the user will be forced to restart the application. This only happens when using Enhanced Navigation and Dashboard screens – forms screens

Solution

This is a known issue for HP – The jar files that drive the connections between the ADF and Forms applications do not work on HP operating systems. This is not an issue for any of the other operating systems that RWMS is certified to run on.

Screen Navigation Delay Using Java 7U85+ Between ADF and Forms Screens

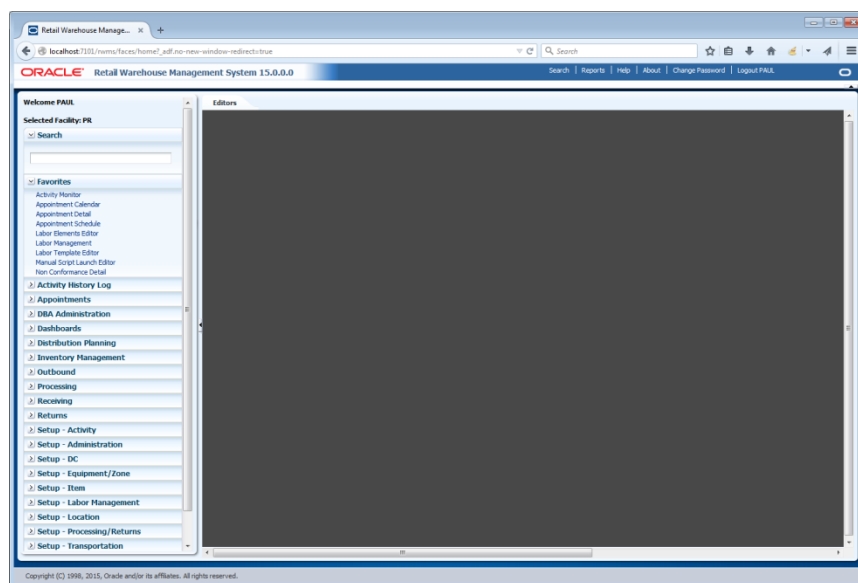
Symptom

Users may experience a delay switching between editors when using Internet Explorer (IE) 11 and JRE versions 7u85-91. The delay typically occurs when a user chooses an ADF based menu option, for example Appointment Calendar, and then chooses a FORMS based menu option, for example Appointment Schedule. The application will switch to the Editors tab but a white area is displayed for a number of seconds before the applet becomes active and the proper Editor appears. This delay appears to be caused by IE's reactivation of the FORMS applet.

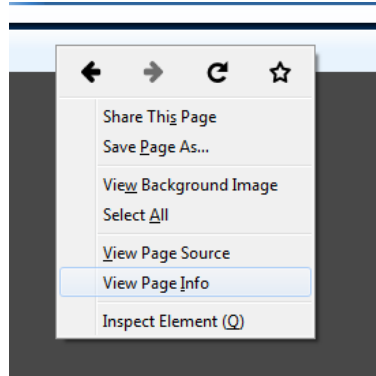
Solution

There is no known fix to IE or the above listed JREs at this time. Users can choose to install JRE 7u80 which does not exhibit the delay.

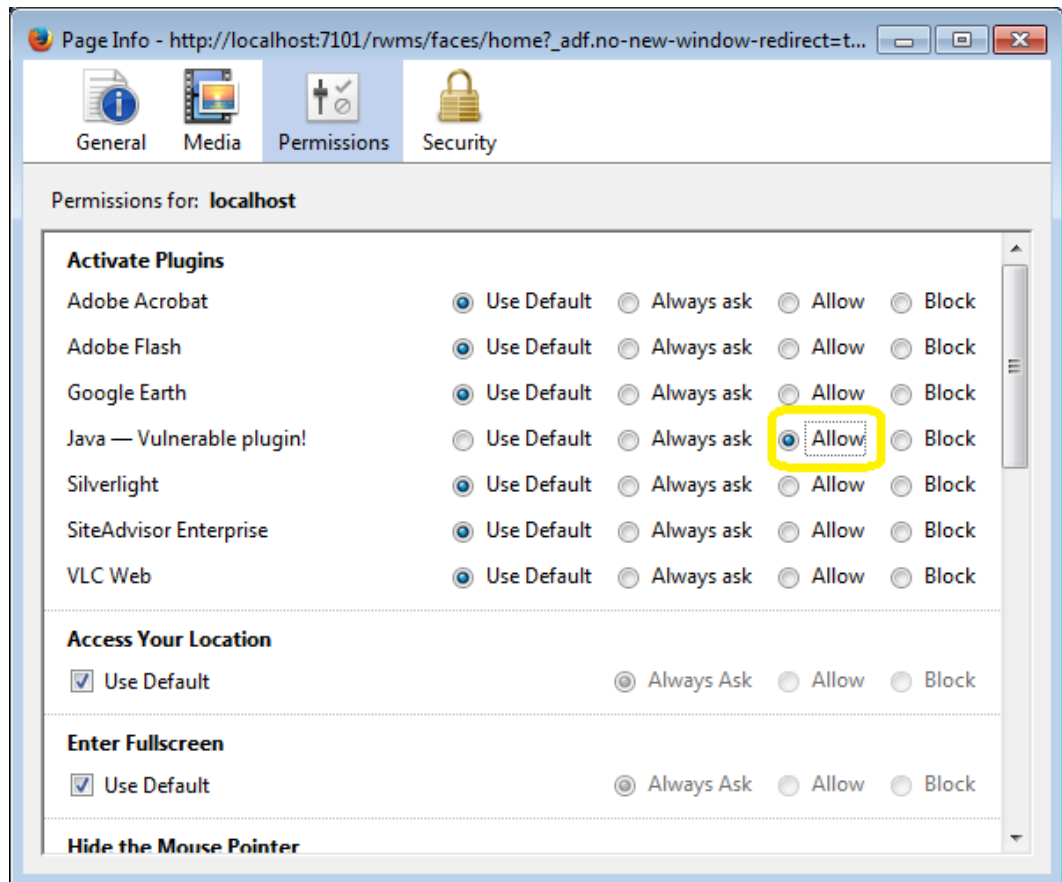
Note: If users are also using Firefox and install JRE 7u80, they must activate the java plugin. If the java plugin is not activated in Firefox, the screen will display as follows:



To activate the 7u80 JRE applet in Firefox, right-click in the grey area and choose View Page Info.



Then click on the Permissions tab and click the radio button to Allow Java.



Appendix: URL Reference

JDBC URL for a Database

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

Thick Client Syntax Non-Pluggable: jdbc:oracle:oci:@<sid>

Thick Client Syntax Pluggable: jdbc:oracle:oci:@<servicename>

<sid>: system identifier for the database

Example: jdbc:oracle:oci:@mysid

Example: jdbc:oracle:oci:@myservicename

Thin Client Syntax Non-Pluggable: jdbc:oracle:thin:@<host>:<port>/<sid>

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

<host>: hostname of the database server

<port>: database listener port

<sid>: non-pluggable system identifier for the database

<servicename>: pluggable system identifier for the database

Example: jdbc:oracle:thin:@myhost:1521/mysid

Example: jdbc:oracle:thin:@myhost:1521:myservicename

JNDI Provider URL for an Application

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

WebLogic:

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

Where,

- <host>: hostname of the WebLogic environment.
- <port>: Managed server port number. This can be found in the <managed server> tag at
<WebLogic_home>/user_projects/domain/<domain_name>/config/config.xml
- <managed_server_name>: This is the managed server name on which the RIB application is deployed.
- <app>: Deployment name for the application.

For example, t3://mspdv161.us.oracle.com:17003/rib-rpm-server/rib-rpm

Note: The JNDI provider URL can have a different format depending on your cluster topology. Consult the WebLogic Server documentation for further details.

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.

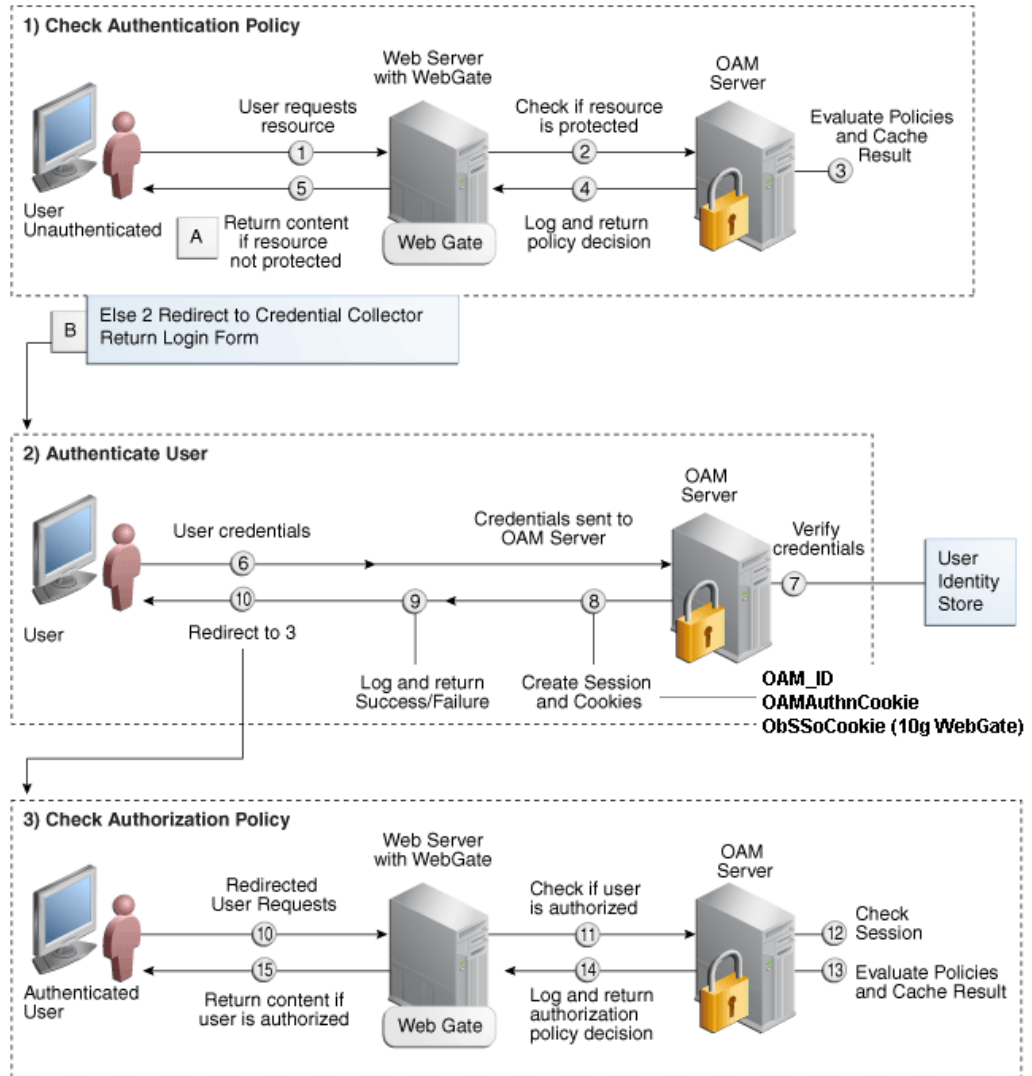
About SSO Login Processing with OAM Agents

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 Guide*11g.

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.

Appendix: Setting Up an Oracle Wallet

The ORACLE Wallet is designed to securely store connection information for an Oracle database, to allow processes to easily and safely connect. This avoids situations where programs would accept a username/password on the command-line (exposing that information to “ps” commands), or storing connection information in plain text configuration files.

This is an OPTIONAL feature. But it is highly advised unless you have a machine where you can ensure the administrator is the only one to sign-on to the server.

After completion of the setup you will be able to enter a connect string, such as sqlplus /@<db alias from tnsname.ora>.

For example:

```
sqlplus /@RWMS150MOCKUSER_polsp02app
```

Note: The Oracle Wallet is setup by the RWMS installer this section is for information on how to create an Oracle Wallet or maintain one that already exists.

Steps to Set up a pl/sql Wallet

1. Create a new directory or navigate to the one created by the RWMS installer:
 - <RETAIL_HOME>/orpatch/rwms_wallet_app
 - a. cd <RETAIL_HOME>/orpatch/
 - b. mkdir rwms_wallet_app
 - c. chmod 755 rwms_wallet_app

Note: By default the permissions on the wallet will allow only the owner to use it, ensuring the connection information is secure. If you are creating a wallet for multiple users you must ensure the permissions are configured to allow only appropriate users to access the wallet.

2. Create a sqlnet.ora in the wallet directory with these contents. It is critical that WALLETS_LOCATION is on line 1 in the file


```
WALLETS_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA =
(DIRECTORY = <RETAIL_HOME>/orpatch/rwms_wallet_app)) )
SQLNET.WALLETS_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```
3. Set up a tnsnames.ora in the wallet directory. This tnsnames.ora will include the standard tnsnames.ora file, and then add a custom entry that is only for use with the wallet (ex: sqlplus /@RWMS150MOCKUSER_polsp02app).

```
RWMS150MOCKUSER_polsp02app =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host =
dbhost)(Port = 1521))) (CONNECT_DATA = (SERVICE_NAME = polsp02app)))
```

Note: It is important not to just copy the tnsnames.ora file, as it quickly becomes out of date.

4. Create the wallet files (initially empty)

- a. Ensure you are in the intended location

```
$ pwd
<RETAIL_HOME>/orpatch/rwms_wallet_app
```

- b. Create the wallet files

```
$ mkstore -wrl . -create
```

- c. Enter password:-> enter your chosen administrative password for the wallet

- d. Enter password again:

Two wallet files are created from the above command:

- ewallet.p12
- cwallet.sso

5. Create the wallet entry that will tie a username/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
RWMS150MOCKUSER_polsp02app wms01user oracle
```

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

```
$ export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
$ export ORACLE_HOME=/scratch/u00/webadmin/product/wls_retail/as_1
$ export PATH=$ORACLE_HOME/bin:$PATH
export
TNS_ADMIN=/vol.rtk/pkg_mocks/rwms150/retail_home_app/orpatch/rwms_wallet_app
$ sqlplus /@RWMS150MOCKUSER_polsp02app
```

```
SQL*Plus: Release 11.1.0.7.0 - Production on Mon Dec 1 14:50:15 2014
```

```
Copyright (c) 1982, 2008, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
```

```
SQL> show user
```

```
USER is "RWMS150MOCKUSER"
```

Additional Wallet Commands

- Delete a credential on wallet:
`mkstore -wrl . -deleteCredential dvols29_wms01user`
- Change the password for a credential on wallet
`mkstore -wrl . -modifyCredential dvols29_wms01user wms01user oracle`
- See what wallet credential entries you have:
`mkstore -wrl . -list`
Returns values like:
`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 value of the entry:
RWMS150MOCKUSER_polsp02app
`mkstore -wrl . -viewEntry oracle.security.client.user1`
Returns value of the entry:
RWMS150MOCKUSER
`mkstore -wrl . -viewEntry oracle.security.client.password1`
Returns value of the entry:
oracle

Once wallet info is set up, create a `wallet_batch_profile` same path location where wallet dir exists. `wallet_batch_profile` should look something like:

```
#source this profile out in order to run batch (run_distribution.sh, etc)
#using wallet entry for schema/password@db rather than the actual password
#so that the command line will not show actual password.
export TNS_ADMIN
=/vol.rtk/pkg_mocks/rwms150/retail_home_app/orpatch/rwms_wallet_appexport
ORACLE_WALLET_ENTRY=RWMS150MOCKUSER_polsp02app
export RDMUSER=''
export RDMPWD='@'$ORACLE_WALLET_ENTRY
```

You will now be able to run a batch command from bin without exporting UP to show schema/password@db. Using wallet, we are able to keep the password invisible.

Setting up Application (Java) Wallet for RWMS .env File

To set up the application (Java) wallet for the RWMS .env file, complete the following steps.

1. Create a new directory called `wallet` under the folder structure through which you run RWMS (for example, `cd <RETAIL_HOME>/extras`).
 - `mkdir javawallet`
2. Navigate to the `save_credential.sh` script to create the java wallet
 - `cd <RETAIL_HOME>/orpatch/deploy/retail-public-security-api/bin`
3. The application wallet requires two entries for BI Publisher reports to work. The first one is `<user_name>/<password>@<servicename>` for RWMS Database. The second one is `username/password` for BI Publisher. See examples in the next step.

4. Create a wallet using the below command:

- `./save_credential.sh -u <USERNAME> -a <ALIASNAME> -p <PARTITION NAME> -l <WALLET PATH>`

Example command for RWMS_DB_CONNECT:

```
./save_credential.sh -u rwms15mockuser -a rwms15mock_polsp09app -p
rwmsinst -l /vol.rtk/pkg_mocks/rwms150/mock/app/retail_home/.javawallet
```

NOTE: It will ask for password [Twice] and the wallet will be created. For this entry the password needs to be in the following format
<db_user_password>@<servicename>

For example: password will be retek@servicename

Example command for RWMS_BI_PWD:

```
./save_credential.sh -u retail.user -a BI_ALIAS -p rwmsinst -l
/vol.rtk/pkg_mocks/rwms150/mock/app/retail_home/.javawallet
```

5. Rerun the above command to put multiple entries into the same wallet.

The above commands shall generate three files as below.

- `cwallet.sso`
- `jazn-data.xml`
- `jps-config.xml`

6. The entries in .env file will look like as shown in example below.

`rwms15inst.env`

```
-----
RWMS_WALLET_PATH=/vol.rtk/pkg_mocks/rwms150/mock/app/retail_home/.javawallet
RWMS_WALLET_PARTITION=rwmsinst
RWMS_WALLET_LOGON=TRUE
RWMS_BI_PWD=BI_ALIAS
RWMS_DB_CONNECT=rwms15mock_polsp09app
```

7. To update the credentials, repeat the steps to create the java wallet and the entries will be over written.

8. To review the wallet to see that the partition and usernames were created correctly:

```
./dump_credentials.sh <RETAIL_HOME>/ .javawallet
```

```
=====
Retail Public Security API Utility
=====
Picked up _JAVA_OPTIONS: -
Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2
=====
Below are the credentials found in the wallet at the
location:/vol.rtk/pkg_mocks/rwms150/mock/app/retail_home/.javawallet
=====

Application level key partition name:rwms15
User Name Alias:rwms15mock_polsp09app User Name:rwms15mockuser
User Name Alias:BI_ALIAS User Name:retail.user
=====
```

9. In order for RWMS to be able to use the Java wallet for the .env files

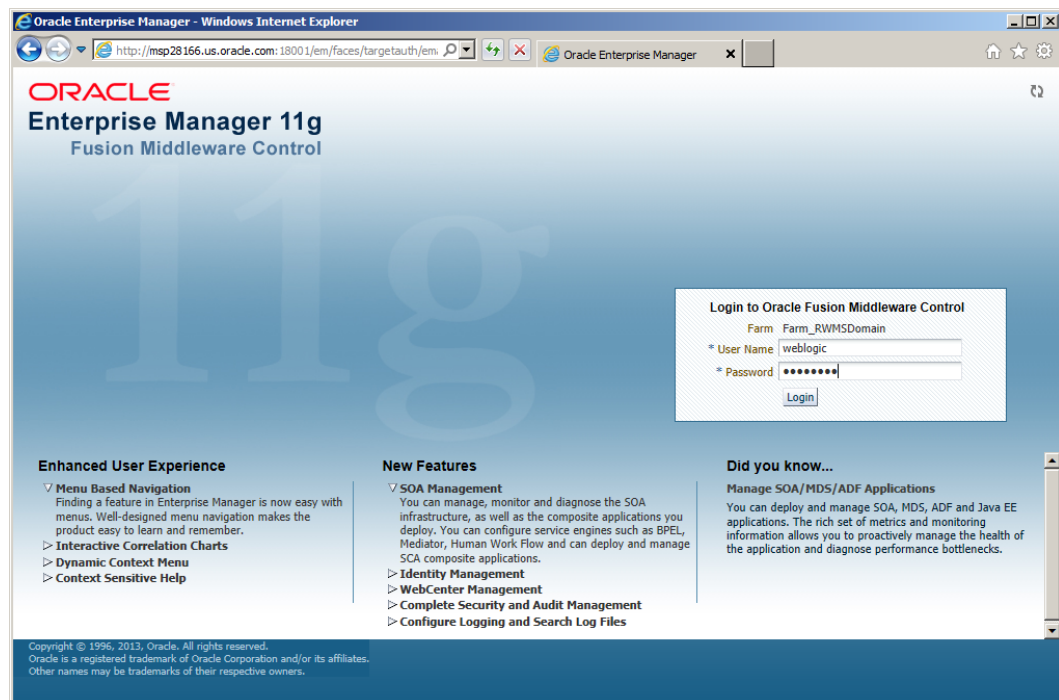
RWMS_DB_CONNECT variable. Validate that the `tnsnames.ora` file has an entry that

can access the database. The location of the tnsnames.ora file will be by the TNS_ADMIN variable in the .env file. The entry should allow the user to log into SQL as follows: sqlplus <username>/<password>l@<servicename>

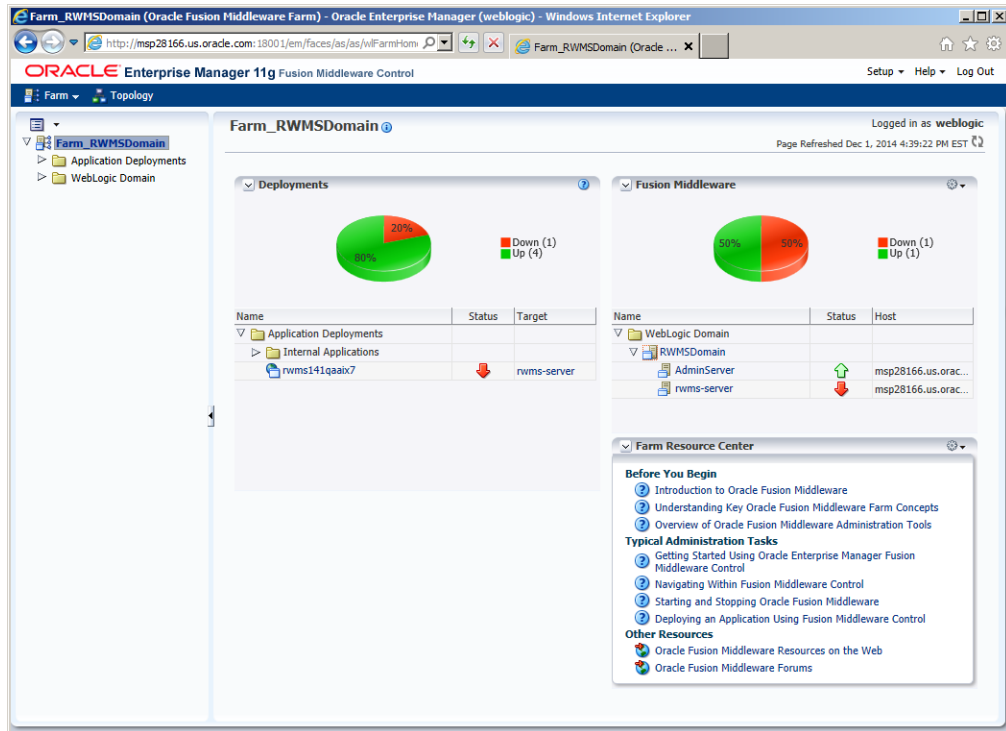
Setting Up WLS Database Credential Store

The WLS Database Credential Store for the RWMS application is done already by the installer. These credentials are setup for the RWMS Manual Script Launch Editor to be able to access the file system and write log files and require a set of credentials that can access the file system. Appendix L section: Database Credential Store Administration goes over how to do this for Oracle Retail Applications using both the WLS Enterprise Manager and WLST commands. Below are specific steps for doing this via the WLS Enterprise manager for RWMS.

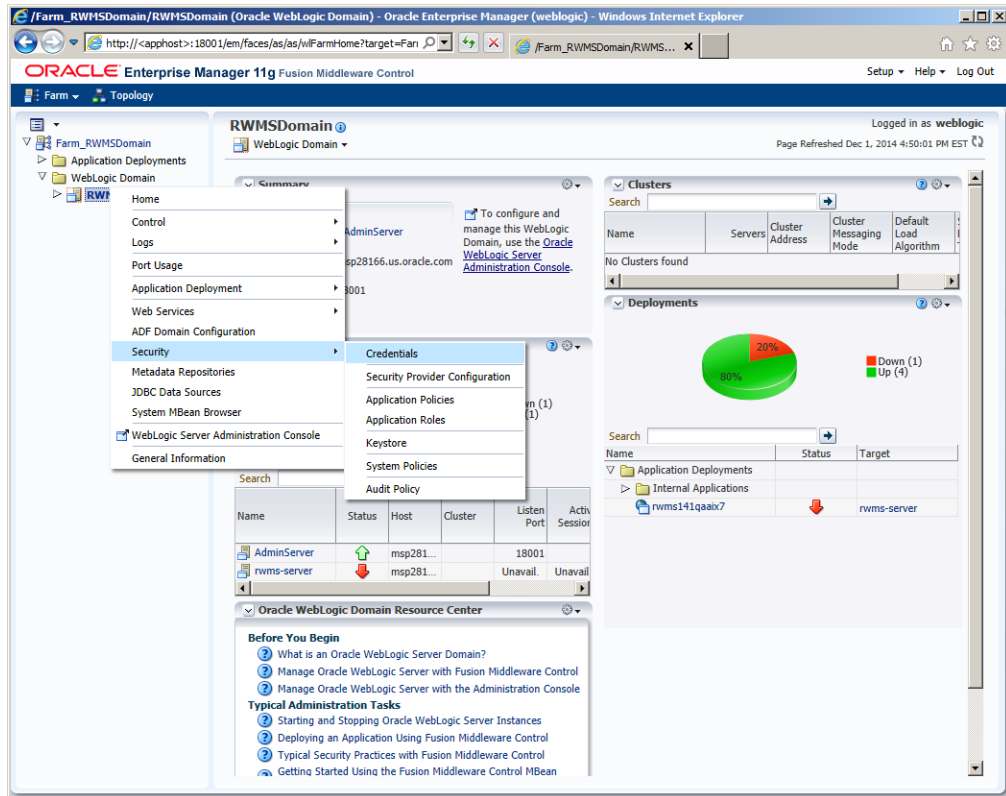
1. Launch WLS Enterprise Manager for the RWMS Enhanced Navigation and Dashboards server:

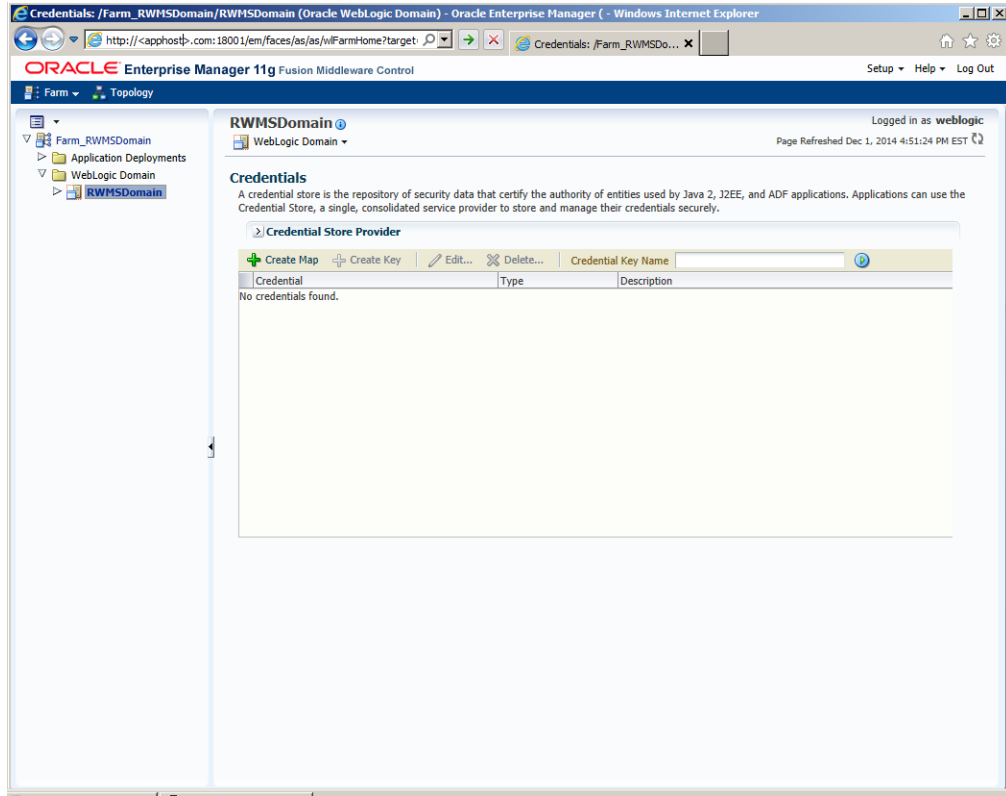


2. Enter your login information.

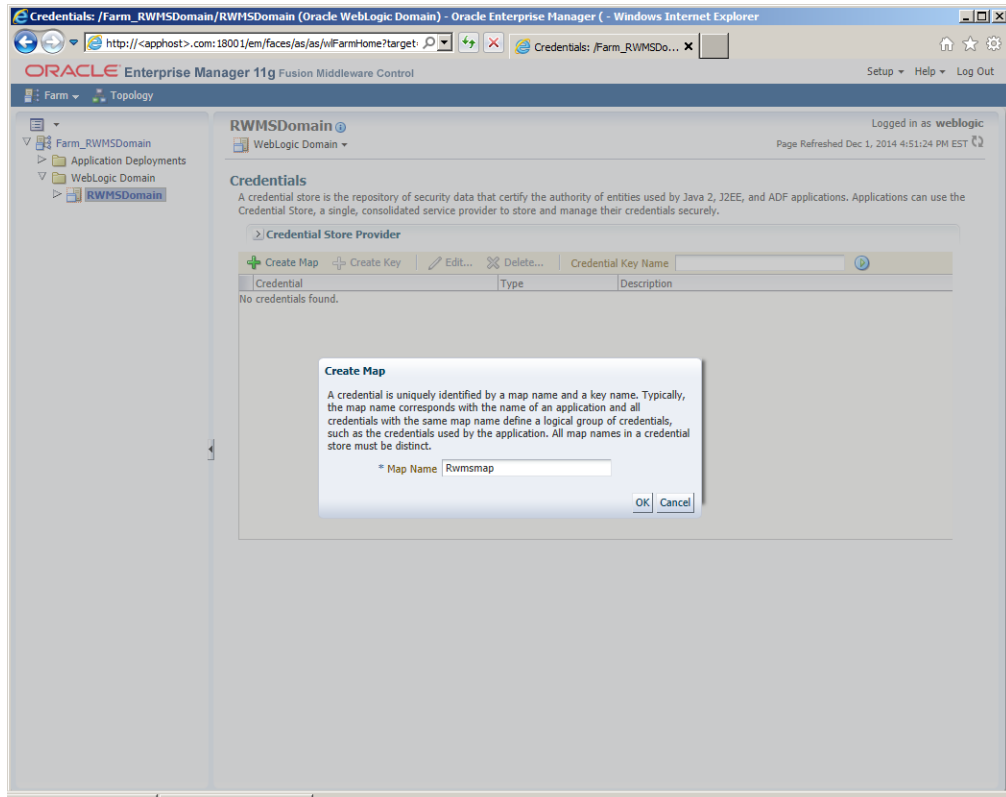


3. Navigate: Weblogic Domain → RWMSDomain in the left navigation pane. Then Using the Weblogic Domain dropdown navigate Security → Credentials.

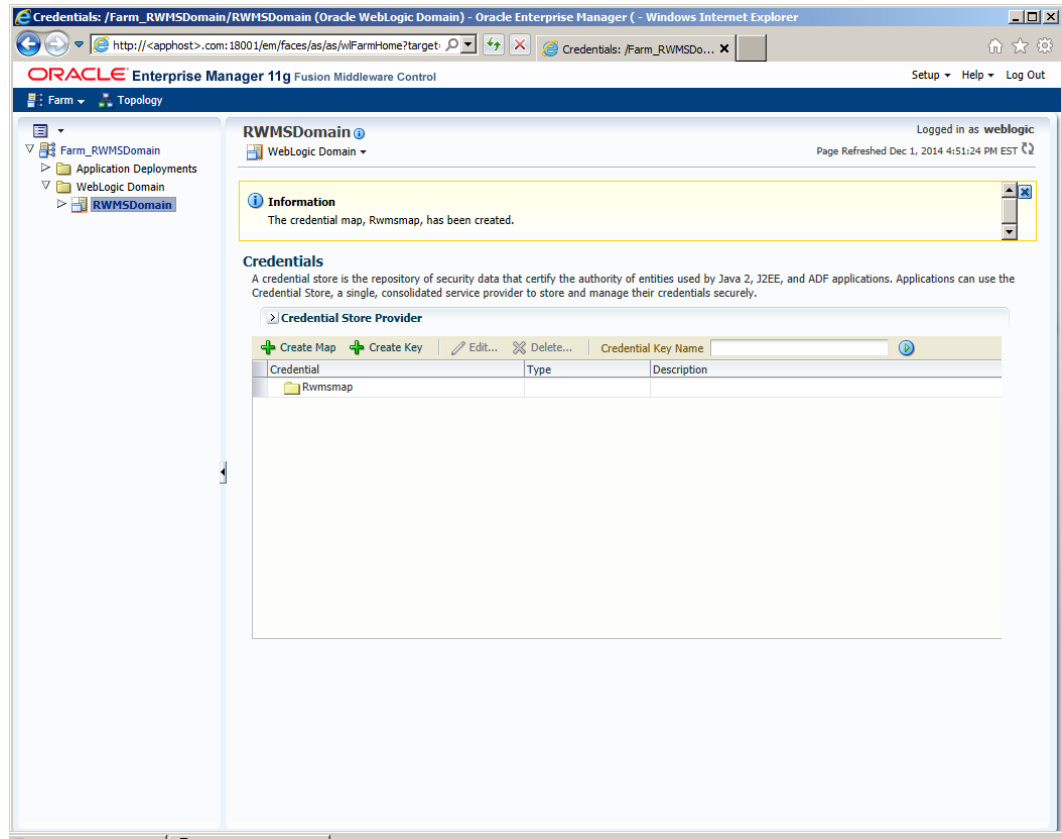




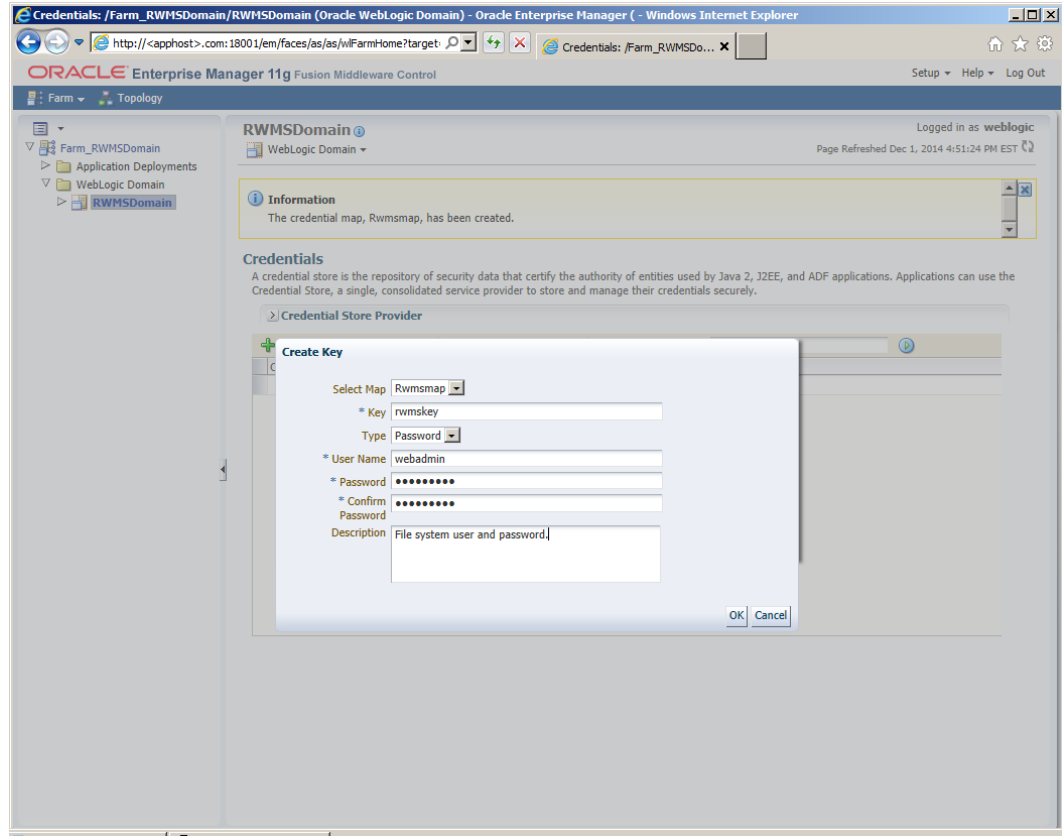
4. Click Create Map.



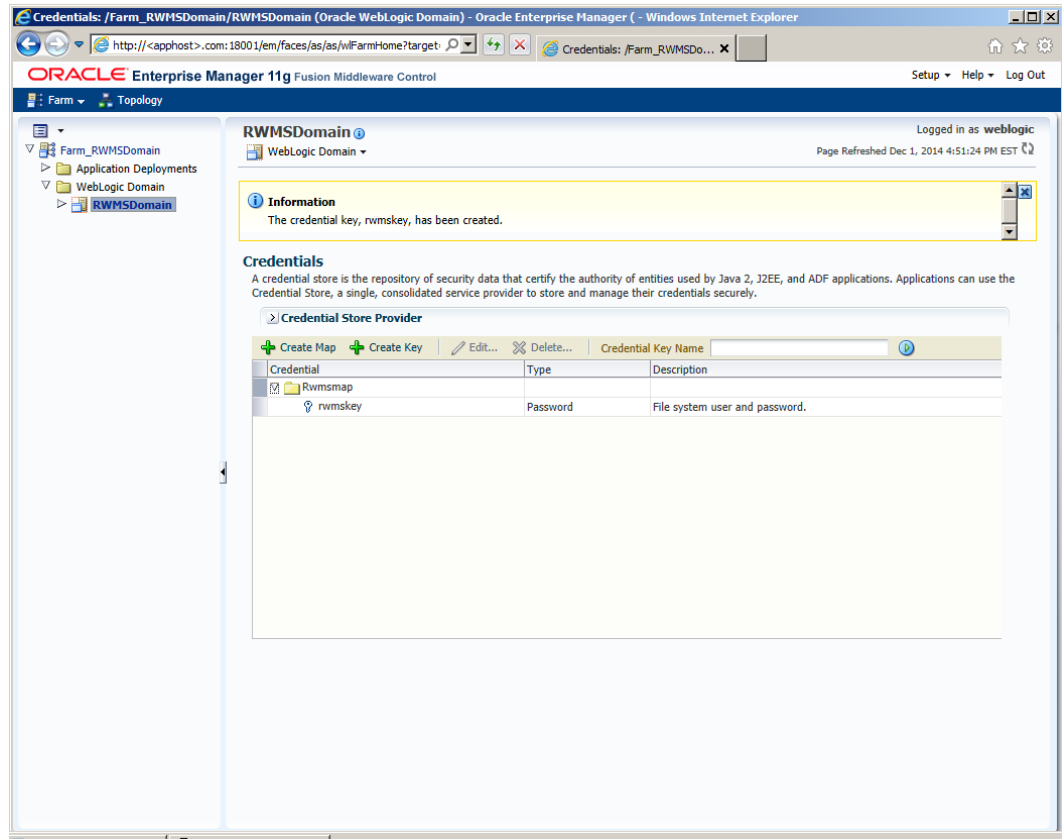
5. Enter in Rwmsmap (This is case sensitive.) and click OK.



6. Click **Create Key** – Enter in the details.
 - Key = rwmskey (This is case sensitive)
 - User Name = <server_user>
 - Password = <server_password>



7. Click OK.



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/rms15/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/rms15/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/12.1.0.2/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/rms15/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/rms15/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_ALIASES should point to the Java wallet entry:
 - `export RETL_WALLET_ALIASES="retl_java_rms01user"`
 - The ORACLE_WALLET_ALIASES should point to the Oracle network wallet entry:
 - `export ORACLE_WALLET_ALIASES="dvols29_rms01user"`
 - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIASES:
 - `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIASES}"`
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/config/domains/wls_retail/REIMDomain/retail/reim15/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/config/domains/wls_retail/REIMDomain/retail/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/reim15/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/config/domains/wls_retail/REIMDomain/retail/reim15/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.
/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15/retail-public-
security-api/bin>
```

dump_credentials.sh

dump_credentials.sh is used to retrieve information from wallet. For each entry found in the wallet, the wallet partition, the alias, and the 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/config/domains/wls_retail/REIMDomain/retail/reim15/config
```

```
Retail Public Security API Utility
```

```
=====
```

```
Below are the credentials found in the wallet at the
location/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15c
onfig
```

```
=====
```

```
Application level key partition name:reim15
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:rms15mock
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/mock15_testing/reim15/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=rms15mock
datasource.credential.alias=RMS-ALIAS
# =====
# ossa related Configuration
#
# These settings are for ossa configuration to store credentials.
# =====

csm.wallet.path=/u00/webadmin/config/domains/wls_retail/REIMDomain/retail/reim15co
nfig
csm.wallet.partition.name=reim15
```

How does the Wallet Relate to Java Batch Program use?

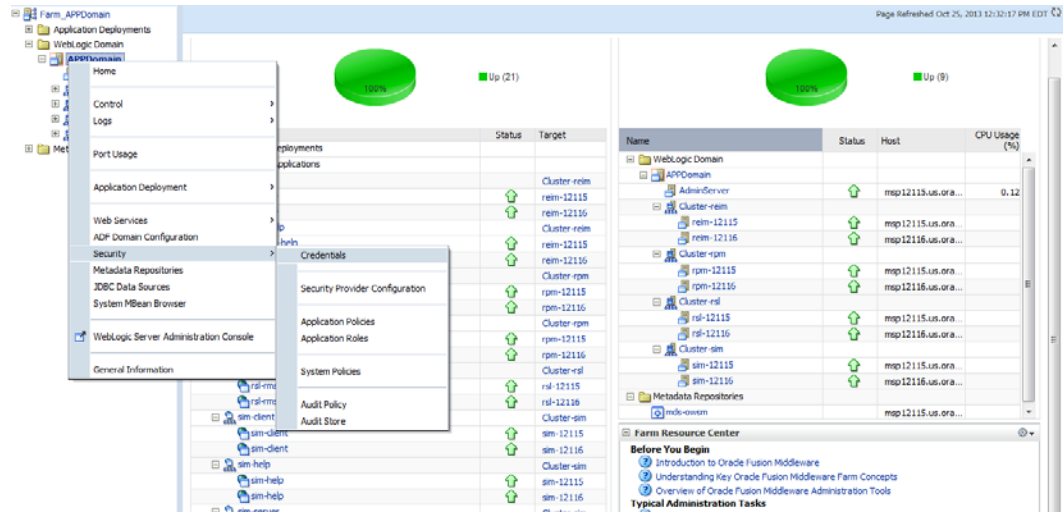
Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser RMS01APP, already on the database. To run a ReIM batch program the format would be: `reimbatchpgmname REIMBAT-ALIAS <other arguments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

Database Credential Store Administration

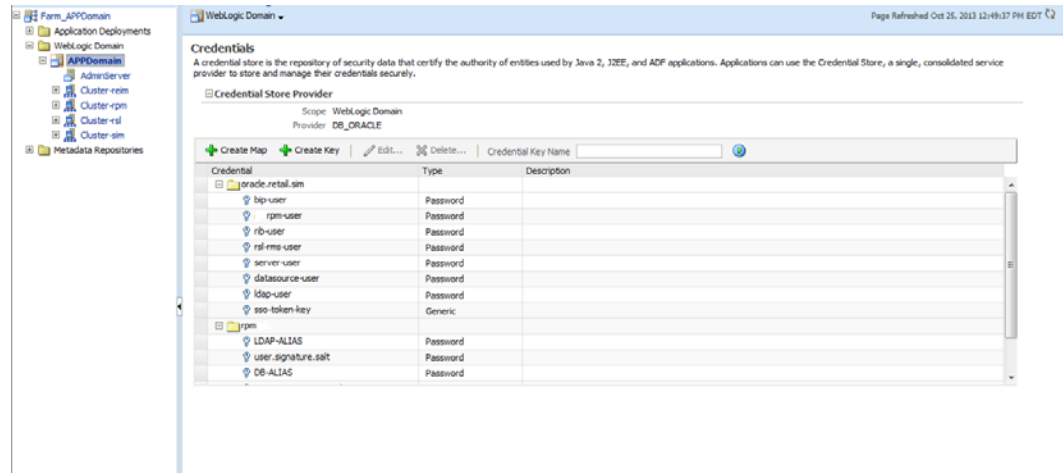
The following section describes a domain level database credential store. This is used in RPM login processing, SIM login processing, RWMS login processing, RESA login processing and Allocation login processing and policy information for application permission. Setting up the database credential store is addressed in the RPM, SIM, RESA, RWMS, and Alloc 15.0 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**

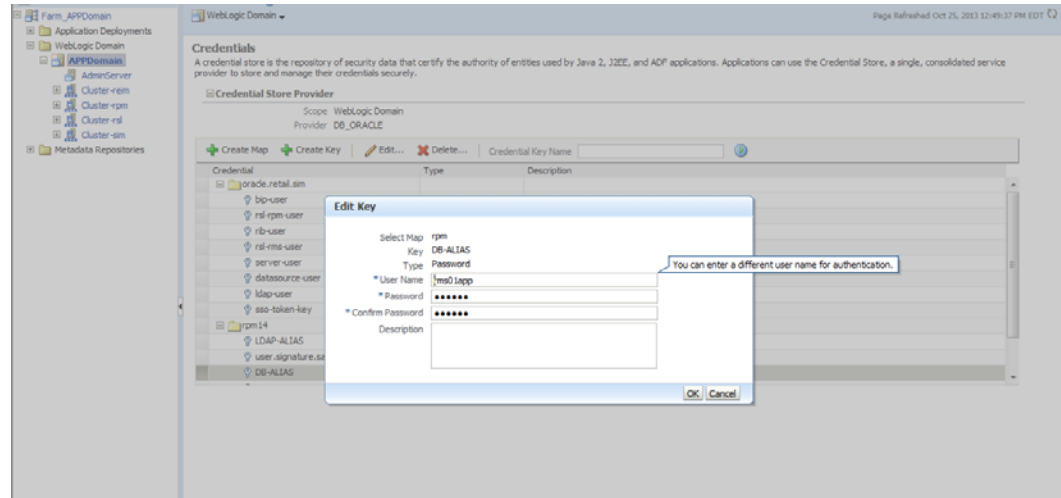


2. Click on Credentials and you will get a screen similar to the following. The following screen is expanded to make it make more sense. From here you can administer credentials.



The Create Map add above is to create a new map with keys under it. A map would usually be an application such as rpm15. 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 (rpm) 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 ReIM , 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.

The weakness with the WSLT/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/fmw
config
```

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

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

Requested Certificates:

User Certificates:

Oracle Secret Store entries:

```
rpm@#3#@DB-ALIAS
rpm@#3#@LDAP-ALIAS
rpm@#3#@RETAIL.USER
rpm@#3#@user.signature.salt
rpm@#3#@user.signature.secretkey
rpm@#3#@WEBLOGIC-ALIAS
rpm@#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 WLS:

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

```
Initializing WebLogic Scripting Tool (WLS)...
```

```
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="rpm",key="DB-ALIAS")
Already in Domain Runtime Tree
```

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

```
*The above means for map rpm15 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	rwms15inst	
			<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
			aip	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	DB credential store		Map=rpm or what you called the app at install time.	Many for app use					<weblogic domain home>/config/fmwc onfig/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
			rpm	<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
			rpm	<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: reim>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name, ex: reim>	<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: reim>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebser vice-alias	
			<installed app name, ex: reim>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			<installed app name, ex: reim>	<LDAP-ALIAS>	cn=REIM.A DMIN,cn=Users,dc=users,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALI AS	
	JAVA	<retail_home>/orpatch/conf/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	reimwebser vice-alias	
			retail_installer	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
			retail_installer	<LDAP-ALIAS>	cn=REIM.A DMIN,cn=Users,dc=users,dc=oracle,dc=com	LDAP user use	Installer	LDAP_ALI AS	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RESA app	DB credential store		Map=resaor 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/ja vaapp_rasm							Each alias must be unique
			retail_installer	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic- alias	
Alloc app	DB credenti al store		Map=alloc or what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwc onfig/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_rasm							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	

Appendix: Manual Forms Compilation

To manually recompile forms, please use orcompile utility

This is only possible after installer has been run and configured Oracle Retail Patch Assistant.

- Set RETAIL_HOME environment variable
- \$RETAIL_HOME/orpatch/bin/orcompile -a RWMS -t FORMS

Usage:

```
orcompile -a <app> -t <type>
```

Potential Apps and Types:

```
RWMS => DB,DB-ADF,FORMS
```

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)
8. Oracle Retail Allocation
9. Oracle Retail Mobile Merchandising (ORMM)
10. Oracle Retail Xstore Office
11. Oracle Retail Xstore Point-of-Service, including Xstore Point-of-Service for Grocery, and including Xstore Mobile
12. Oracle Retail Xstore Environment
13. Oracle Retail EFTLink
14. Oracle Retail Store Inventory Management (SIM), including Mobile SIM
15. Oracle Retail Predictive Application Server (RPAS)
16. Oracle Retail Batch Script Architecture (BSA)
17. Oracle Retail Demand Forecasting (RDF)
18. Oracle Retail Category Management Planning and Optimization/Macro Space Optimization (CMPO/MSO)
19. Oracle Retail Replenishment Optimization (RO)
20. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
21. Oracle Retail Regular Price Optimization (RPO)
22. Oracle Retail Merchandise Financial Planning (MFP)
23. Oracle Retail Size Profile Optimization (SPO)
24. Oracle Retail Assortment Planning (AP)
25. Oracle Retail Item Planning (IP)
26. Oracle Retail Item Planning Configured for COE (IP COE)
27. Oracle Retail Advanced Inventory Planning (AIP)
28. Oracle Retail Integration Bus (RIB)
29. Oracle Retail Services Backbone (RSB)

- 30.** Oracle Retail Financial Integration (ORFI)
- 31.** Oracle Retail Data Extractor for Merchandising
- 32.** Oracle Retail Clearance Optimization Engine (COE)
- 33.** Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
- 34.** Oracle Retail Insights, including Retail Merchandising Insights (previously Retail Merchandising Analytics) and Retail Customer Insights (previously Retail Customer Analytics)