Oracle® Retail Warehouse Management System

Installation Guide

Release 14.0

E50617-03

November 2014

Oracle® Retail Warehouse Management System Installation Guide, Release 14.0

Primary Author: Nathan Young

Contributors: Amandeep Bhatti

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 ix

Preface xi

Audience xi

Related Documents xi

Customer Support xi

Review Patch Documentation xi

Improved Process for Oracle Retail Documentation Corrections xii

Oracle Retail Documentation on the Oracle Technology Network xii

Conventions xii

1 Preinstallation Tasks 1

Check Supported Database Server Requirements 1

Check Supported Application Server Requirements 2

Check Supported Web Browser and Client Requirements 3

Supported Radio Frequency Device Requirements 4

Supported Oracle Retail Products 4

UNIX User Account Privileges to Install the Software 4

Create Staging Directory for RWMS Installer 5

2 RAC and Clustering 7

3 Database Installation Tasks 9

Create the RWMS Database 9

Create the Database Instance Using a Template 9

Create the required RWMS Tablespaces 10

Create the Schema Owner for RWMS 12

Run the RWMS Database Schema Installation 13

4 Application Server Installation Tasks-RWMS Forms 15

Prepare Application Server for RWMS 16

Install RWMS Oracle Forms 16

5 Enhanced Navigation and Dashboards Installation 19

Steps to Create the New Domain with ADF Libraries 20

Start the RWMSDomain and the Node Manager 28

Start the Managed Server 29

Run the RWMS Installer 29

Custom Authentication Provider set up 30

Disable User Lock Out Settings 33

Application: Logging File Settings 33

Set the Class Path for the Managed Server 34

6 Manual Installation and Configuration Tasks 35

This section covers the installation and configuration of external components and other dependencies for RWMS 14.0 35

Prerequisites 35
Path References Used in this Section....35
BI Server Component Installation Tasks....36
Installation Process Overview....37
Install Oracle BI EE 11g....37
Installing the RWMS BI Publisher Templates....55
Configuring the RWMS JDBC connection....56
Installing WebUtil Support Libraries....57
Configuring Forms to use BI and WebUtil....58
Check the Configuration of formsweb.cfg....58
Configuring the Forms Installation Environment Files....60
Adding Radio Frequency Launch Configurations....66
File Placement....66
Configure .env file Variables....66
Update the RF Launch Modules....67
8 Web Services Installation 69
Create a Managed Server....69
Create a Datasource....69
Deploy RWMS Service EAR File....69
Configure Web Service Security....71
9 RWMS Patching Procedures 73
Patching Changes in RWMS 14.0....73
Patch Concepts....73
Patch Structure....74
Patching Utility Overview....74
Patching Considerations....75
Patch Types....75
Version Tracking....75
Environment Configuration....75
Retained Database Scripts....75
Reloading Content....76
Backups....76
Disk Space....76
Maintenance Considerations....76
Apply all Patches with Installer or Oracle Retail Patch Assistant....76
Password Changes....77
Infrastructure Directory Changes....77
DBManifest Table....77
MMHOME relationship to Database....77
Patching Operations....77
Running Oracle Retail Patch Assistant....77
Merging Patches....85

Compiling Application Components 86
Customization 87
Patching Considerations with Customized RWMS Files and Objects 87
Registering Customized Files 88
Extending Oracle Retail Patch Assistant with Custom Hooks 90
Troubleshooting Patching 92
ORPatch Log Files 92
Restarting ORPatch 92
Manual DBManifest Updates 93
Manual Restart State File Updates 93
DISPLAY Settings When Compiling Forms 93
Patching Prior to First Install 94
Providing Metadata to Oracle Support 95
A Appendix: Oracle Database 11g Parameter File 97
B Appendix: Tablespace Creation Scripts 99
Note: If you have created tablespaces during the implementation or upgrade of the Retail Merchandising System 14, you no longer require to create another set of tablespaces for the Warehouse Management System as they use the same ones. 99
Non-Encrypted Tablespace Creation 99
Encrypted Tablespace Creation 99
Configure a Wallet: 100
Encryption at Tablespace Level 100
C Appendix: Sample Oracle Net Files for the Server 101
listener.ora 101
tnsnames.ora 102
D Appendix: RWMS Database Schema Installation Screens 103
E Appendix: RWMS Application(forms) Installation Screens 113
F Appendix: RWMS Enhanced Navigation and Dashboards Installation Screens127
G Appendix: Common Installation Errors 145
Database Installer Hangs on Startup 145
Warning: Could not create system preferences directory 145
WLS_FORMS server failed to restart 146
Warning: Couldn't find X Input Context 146
Unresponsive Drop-Downs 147
Could not execl robot child process: Permission denied 148
ConcurrentModificationException in Installer GUI 148
ORA-04031 (unable to allocate memory) error during database schema installation 148
X Error of failed request: BadWindow (invalid Window parameter) 149
RIB Errors 149
Error Connecting to Database URL 150

H Appendix: URL Reference 151
JDBC URL for a Database 151
JNDI Provider URL for an Application 151
I Appendix: Single Sign-On for WebLogic 153
What Do I Need for Single Sign-On? 153
Can Oracle Access Manager Work with Other SSO Implementations? 153
Oracle Single Sign-on Terms and Definitions 154
What Single Sign-On is not 155
How Oracle Single Sign-On Works 155
Installation Overview 157
User Management 157
J Appendix: Setting Up an Oracle Wallet 159
Steps to Set up a pl/sql Wallet 159
Additional Wallet Commands 161
Setting up Application (Java) Wallet for RWMS .env File 161
K Appendix: Setting Up Password Stores with wallets/credential stores 163
About Database Password Stores and Oracle Wallet 163
Setting Up Password Stores for Database User Accounts 164
Setting up Wallets for Database User Accounts 165
For RMS, RWMS, RPM Batch using sqlplus or sqlldr, RETL, RMS, RWMS, and ARI 165
Setting up RETL Wallets 167
For Java Applications (SIM, ReIM, RPM, RIB, RSL, AIP, Alloc batch, RETL) 168
How does the Wallet Relate to the Application? 171
How does the Wallet Relate to Java Batch Program use? 171
Database Credential Store Administration 171
Managing Credentials with WSLT/OPSS Scripts 175
listCred 176
updateCred 177
createCred 177
deleteCred 177
modifyBootStrapCredential 178
addBootStrapCredential 179
Quick Guide for Retail Password Stores (db wallet, java wallet, DB credential stores) 181
L Appendix: Manual Forms Compilation 189
M Appendix: Web Browser Configuration 191
Mozilla Firefox Version 17.0 191
N Appendix: Installation Order 193
Enterprise Installation Order 193

Send Us Your Comments

Oracle Retail Warehouse Management System, Installation Guide, Release 14.0

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

Related Documents

For more information, see the following documents in the Oracle Retail Warehouse Management System Release 14.0 documentation set:

- *Oracle Retail Warehouse Management System Data Model*
- *Oracle Retail Warehouse Management System User Guide*
- *Oracle Retail Warehouse Management System Online Help*
- *Oracle Retail Warehouse Management System Operations Guide*
- *Oracle Retail Warehouse Management System Radio Frequency User Guide*
- *Oracle Retail Warehouse Management System Release Note*
- *Oracle Retail Warehouse Management System Security Guide*
- *Oracle Retail Enterprise Integration Guide*

See also:

- Oracle Retail Integration Bus documentation

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 14.0) or a later patch release (for example, 14.0.1). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

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

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

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

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

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

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

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

Conventions

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

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

1 Preinstallation Tasks

This chapter includes steps to complete before installation.

Check Supported Database Server Requirements

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

Supported on	Versions Supported
Database Server OS	OS certified with Oracle Database 11gR2 Enterprise Edition. Options are: ▪ Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.3) with the following specifications: **Components:** ▪ Oracle Partitioning ▪ Examples CD (Formerly the companion CD) **Oneoff Patches:** ▪ 13036331: ORA-01031 INSUFFICIENT PRIVILEGES WHEN GRANTING EXECUTE ON DBMS PACKAGES. **Other components:** ▪ Perl 5.0 or later ▪ X-Windows interface

Note: A working X-Server must be defined even when working with the text versions of the installer. It must be configured to accept XClients from the installation server in its access control list. Refer to XServer documentation for more information on using the xhost or equivalent commands.

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 11g Release1 (11.1.2.1). Options are: ▪ Oracle Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 6 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 7.1 (Actual hardware or LPARs) ▪ Solaris 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Application Server	Oracle Fusion Middleware 11g Release 2 (11.1.2.1) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.6) ▪ Oracle Forms Services 11g Release 2 (11.1.2.1) **Oneoff Patches:** ▪ Patch 17448420: MANIFEST ATTRIBUTE ERROR IN JAVA CONSOLE WHILE RUNNING FORMS URL WITH 7U45_B11 **Java:** ▪ JDK 1.7+ 64 bit **For Enhanced Navigation and Dashboards** **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.6) ▪ Oracle ADF (Application Development Framework) 11.1.1.6 ▪ Enterprise Manager Application 11.1.1.6 **Optional (if SSO is required)** ▪ Oracle WebTier 11g (11.1.1.6) Oracle Access Manager 11g Release 1 (11.1.1.5) Note: A separate WebLogic 10.3.5 installation is required for Oracle Access Manager 11g. ▪ Oracle Access Manager Agent (WebGate) 11g Release 1 (11.1.1.5) ▪ Oracle Identity Management 11gR1 (11.1.1.6) **Other components:** ▪ Oracle BI Publisher 11g (11.1.1.6.8)

Check Supported Web Browser and Client Requirements

General requirements for client running RWMS include the following.

Requirement	Version
Operating system	Windows XP or Windows 7
Terminal Server	Windows Server 2008
Display resolution	1024x768 or higher
Processor	2.6GHz or higher
Memory	1GByte or higher
Networking	intranet with at least 10Mbps data rate
Sun Java Runtime Environment	1.7.0_21+
Browser	Microsoft Internet Explorer 8 or 9 Mozilla Firefox 17.0

Note: Windows XP is supported until its End-of-Life (April 2014)

See Appendix: Web Browser Configuration for more information on configuring Web browsers with Java Runtime Environment versions.

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)	14.0
SIM	14.0

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.

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 rwms14installer.zip file from the RWMS 14.0 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 rwms14installer.zip file. This creates an rwms/installer/ subdirectory under STAGING_DIR.

2 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 an 11.2.0.3 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 11.2.0.3 Oracle Internet Directory database with the WebLogic 10.3.6 cluster. It is suggested that a Web Tier 11.1.1.6 installation be configured to reflect all application server installations if SSO will be utilized.

References for Configuration:

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

3

Database Installation Tasks

If a database has been created for the Retail Merchandising System 14 (RMS) and you would like the Retail Warehouse Management System (RWMS) to be in the same database, you can skip this step and advance through this install guide to the "Create the Schema Owner for RWMS" section. However, if you wish that RWMS resides in a dedicated database, follow the steps below to create a new one.

Create the RWMS Database

It is assumed that Oracle Enterprise Edition 11g Release 2, 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 11g 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 a Template

Create a database using characterset AL32UTF8

Note: Prior to Release 14.0, the characterset was UTF8).

Prerequisites:

1. The 11.2.0.3 rdbms binary must have already been installed.
2. Depending on the OS platform the database is built on, copy one of these files from STAGING_DIR/rwms/installer/create_db into the $ORACLE_HOME/assistants/dbca/templates directory and unzip it.

Template Name	OS
Retail_DB_Template_11.2.0.3_AIX_Release.tar.gz	AIX
Retail_DB_Template_11.2.0.3_Linux_x86_64_Release.tar.gz	Linux
Retail_DB_Template_11.2.0.3_Solaris_Release.tar.gz	Solaris (SPARC)
Retail_DB_Template_11.2.0.3_HPIA64_Release.tar.gz	HP-UX Itanium

Note: The remainder of the example uses a Linux example.

3. Unzip and untar the file you have copied so the 4 files below are present in the directory

```
$ gzip -dc Retail_DB_Template_11.2.0.3_Linux_x86_64_Release.tar.gz |tar -xvf –
```

4. After the unzip/untar, these 4 files should be present and you are ready to create the database instance.
   - Retail_DB_Template_11.2.0.3_Linux_x86_64_Release.ctl
   - Retail_DB_Template_11.2.0.3_Linux_x86_64_Release.dbc
   - Retail_DB_Template_11.2.0.3_Linux_x86_64_Release.dfb
   - Retail_DB_Template_11.2.0.3_Linux_x86_64_Release_variables.txt

Instance Creation in Silent Mode Using DBCA

1. Set values ORACLE_HOME for ORACLE_BASE.

```
$>  export ORACLE_HOME=<full_path_of_ORACLE_HOME>
$>  export ORACLE_BASE=<path_of_ORACLE_BASE>
```

2. Update/set the values of ORA_DATA1 and ORA_DATA2 in the file $ORACLE_HOME/assistants/dbca/templates/Retail_DB_Template_11.2.0.3_Linux_x86_64_Release_variables.txt so they hold the paths of the datafile locations as seen in the example below:

```
ORA_DATA1=/u02/oradata
ORA_DATA2=/u03/oradata
```

3. Execute dbca in silent mode to create the database with characterset using AL32UTF8.

```
cd $ORACLE_HOME/bin

./dbca -silent -createDatabase -templateName
Retail_DB_Template_11.2.0.3_Linux_x86_64_Release.dbc -gdbname <DB NAME> -sid
<DB NAME> -characterSet AL32UTF8 -nationalCharacterSet AL16UTF16 -sysPassword
<SYS PASSWORD> -systemPassword <SYSTEM PASSWORD> -emConfiguration NONE -
variablesFile
$ORACLE_HOME/assistants/dbca/templates/Retail_DB_Template_11.2.0.3_
OS_Platform_Release_variables.txt -continueOnNonFatalErrors true
```

   Substitute <DB NAME> , <SYS PASSWORD> and <SYSTEM PASSWORD>with appropriate values.

4. A database instance is created.
5. A corresponding Pfile has also been generated. Adjust the values accordingly.
6. Configure a listener and create the tnsnames entry.

Create the required RWMS Tablespaces

Release 14 replaces the tablespaces RETEK_DATA and RETEK_INDEX with new tablespaces RETAIL_DATA and RETAIL_INDEX. In addition, release 14.0 introduces two new tablespaces, ENCRYPTED_RETAIL_DATA and ENCRYPTED_RETAIL_INDEX.

> **Note:** If you have created tablespaces during the implementation or upgrade of the Retail Merchandising System 14, 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:
   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.
5. If you hold an Advanced Security Option license and wish to use TDE tablespace encryption:
   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.
   4. Refer to Appendix B 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
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 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 RWMS14DEV.
   - Password – the password associate with RWMS14DEV.
   - Temp Tablespace – the temporary tablespace for RWMS14DEV.
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. The settings from the RWMS 14.0 full install might be in this file, and running the installer again for the patch clears out some of the settings that are not used by the installer's patch mode.
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

   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
> ```

7. 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/rwms14/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>

4

Application Server Installation Tasks-RWMS Forms

It is assumed that WebLogic 11g version 10.3.6 (WLS) with Forms 11.1.2.1 has already been installed. If not, refer to "Check Application Server Requirements" in Chapter 1, "Preinstallation Tasks" before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in "Create Staging Directory for RWMS Application Files" in Chapter 1.

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 6.

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.

A copy tnsnames.ora file must be created for the $ORACLE_INSTANCE/config location. If the file is not copied to this location, forms will not compile correctly. See Appendix: Tablespace Creation Script B for an example setup of the tnsnames.ora file.

Note: It is necessary to create Domain directory under WEBLOGIC_HOME/user_projects/domains/. If this is not where it is located, the installer will not be able to copy the files from the post directory over to the WebLogic Domain and this should be done manually.

Prepare Application Server for RWMS

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

> **Note:** ORACLE_HOME is the location where Oracle Forms 11gR2 has been installed.
>
> ORACLE_INSTANCE is the instance that is created during installation of Oracle Forms 11gR2 and contains the executables to compile forms.

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

   $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
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. Uncomment it if it is already present.

   ```
   Tk2Motif*fontMapCs: iso8859-2=AL32UTF8
   ```
4. Copy $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/Tk2Motif.rgb to $ORACLE_HOME/guicommon/tk/admin/Tk2Motif.rgb.

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. This directory was created when the rwms14installer.zip file was expanded under STAGING_DIR.
3. Set and export the following environment variables.

Variable	Description	Example
DOMAIN_HOME	The location of the WebLogic domain.	DOMAIN_HOME=/u00/webadmin/product/10.3.X/WLS/ user_projects/domains/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 where the RWMS schema resides	ORACLE_SID=rwmsdb

Variable	Description	Example
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/jdk1.7.0_45.64bit

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

   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 INSTALL_DIR for the remainder of this chapter. Do not provide an INSTALL_DIR that is located at or underneath STAGING_DIR.

5. After the installation is complete, you can check its log file: STAGING_DIR/rwms/installer/logs/rwms-install.<timestamp>.log. The directory INSTALL_DIR/orpatch/logs/detail_logs/oraforms_rwms/ will contain log files for specific forms, with information about possible failed compilations.

6. 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`

7. If during the screens you chose not to have the installer automatically configure WebLogic, after the installation is complete, follow the post installation tasks by making backups of the listed files and copying the required files to the specified location.

Example

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

/u00/webadmin/product/10.3.X/WLS/user_projects/domains/ClassicDomain/config/fmwcon
fig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/forms/registry/oracle/fo
rms/registry/Registry.dat
/u00/webadmin/product/10.3.X/WLS/user_projects/domains/ClassicDomain/config/fmwcon
fig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/formsweb.cfg

Have the Weblogic administrator stop WLS_FORMS,
copy everything in /u00/webadmin/WADE_RWMS14/mmhome/post
to /u00/webadmin/product/10.3.X/WLS to update the files
and then start WLS_FORMS
for the changes to take effect.

example: cp -R * /u00/webadmin/product/10.3.X/WLS
```

5

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 11g Release 1 (10.3.6). 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 11.1.1.6 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.

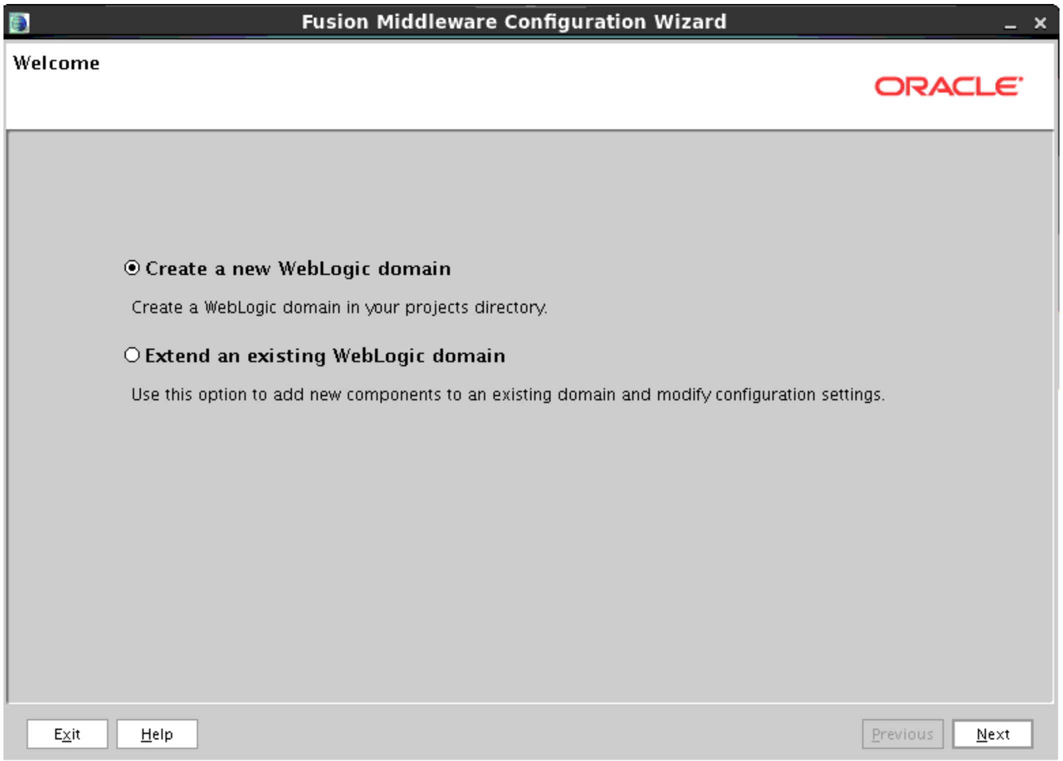
You can create a new weblogic server and install ADF libraries on it or create a weblogic domain on the weblogic server that has the ClassicDomain deployed

Steps to Create the New Domain with ADF Libraries

1. Set the required environment variables

   ```
   export JAVA_HOME=<JDK_HOME>
   export PATH=$JAVA_HOME/bin:$PATH
   ```

2. Change directories to $ORACLE_HOME/wlserver_10.3/common/bin and run the config.sh scripts to create the new RWMSDomain and Enterprise Manager.
3. Select **Create a new WebLogic domain** and click **Next**.



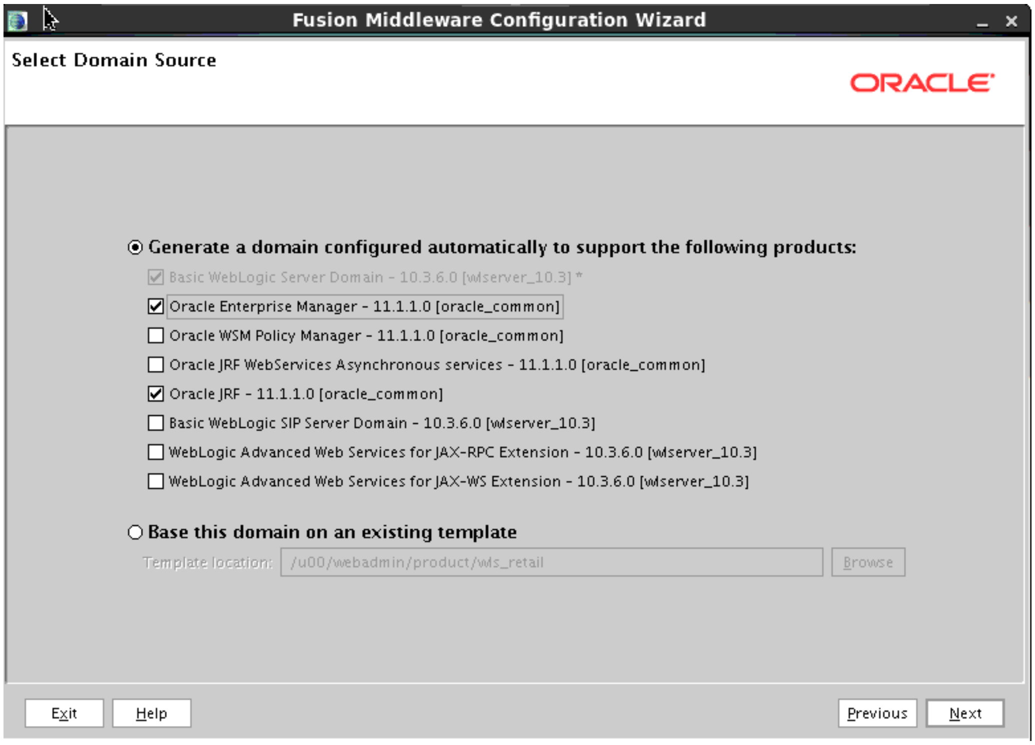
● Create a new WebLogic domain

Create a WebLogic domain in your projects directory.

○ Extend an existing WebLogic domain

Use this option to add new components to an existing domain and modify configuration settings.

4. Select **Oracle JRF** and the **Oracle Enterprise Manager** and click **Next**.



Select Domain Source

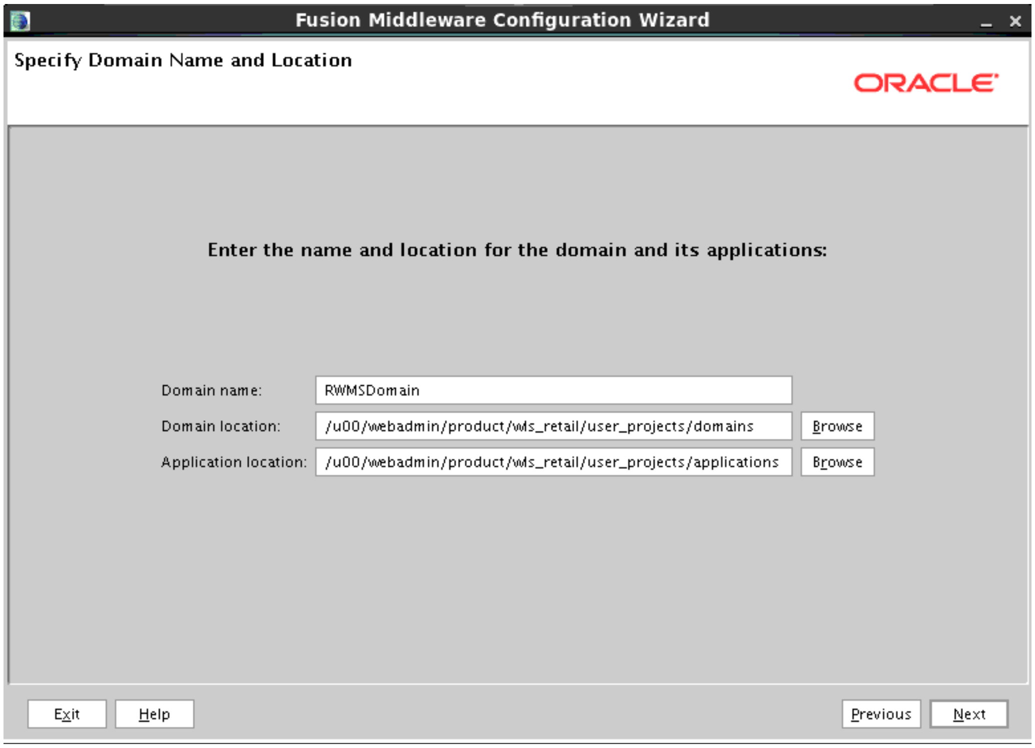
● Generate a domain configured automatically to support the following products:

- ☑ Basic WebLogic Server Domain - 10.3.6.0 [wlserver_10.3] *
- ☑ Oracle Enterprise Manager - 11.1.1.0 [oracle_common]
- ☐ Oracle WSM Policy Manager - 11.1.1.0 [oracle_common]
- ☐ Oracle JRF WebServices Asynchronous services - 11.1.1.0 [oracle_common]
- ☑ Oracle JRF - 11.1.1.0 [oracle_common]
- ☐ Basic WebLogic SIP Server Domain - 10.3.6.0 [wlserver_10.3]
- ☐ WebLogic Advanced Web Services for JAX-RPC Extension - 10.3.6.0 [wlserver_10.3]
- ☐ WebLogic Advanced Web Services for JAX-WS Extension - 10.3.6.0 [wlserver_10.3]

○ Base this domain on an existing template

Template location: /u00/webadmin/product/wls_retail Browse

5. Change the Domain name from the default and click **Next**. For example: RWMSDomain.



Specify Domain Name and Location

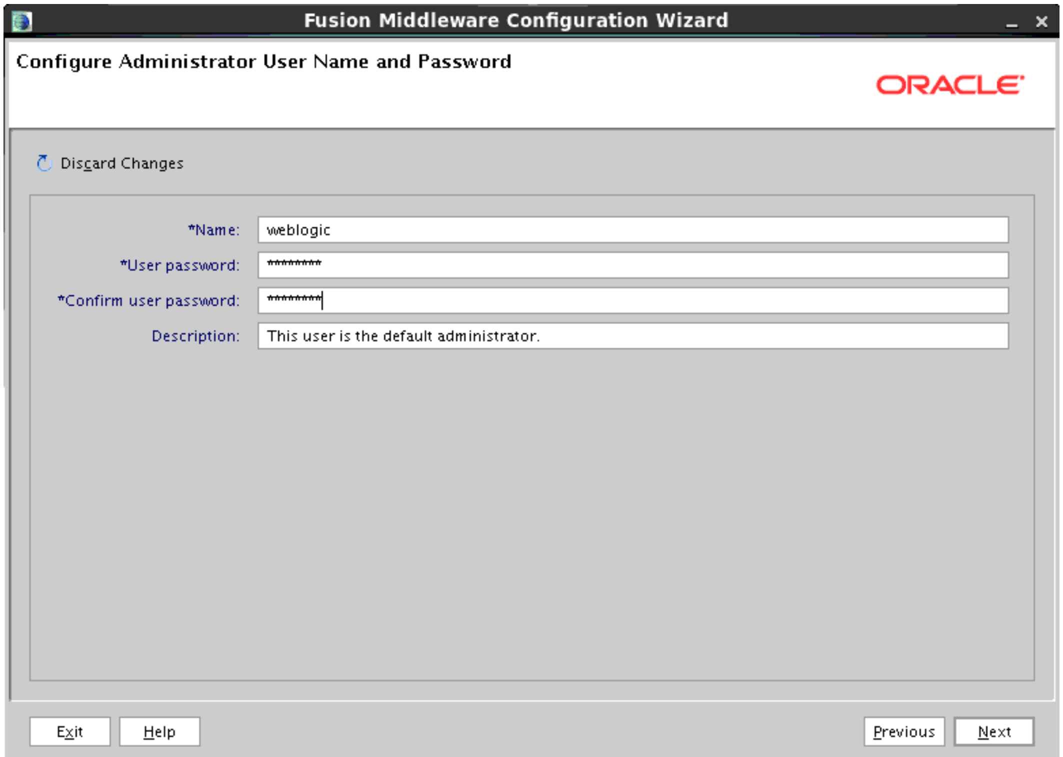
Enter the name and location for the domain and its applications:

Domain name: RWMSDomain

Domain location: /u00/webadmin/product/wls_retail/user_projects/domains Browse

Application location: /u00/webadmin/product/wls_retail/user_projects/applications Browse

6. Enter in the credentials used for the WebLogic domains AdminServer and click **Next**.



Fusion Middleware Configuration Wizard

Configure Administrator User Name and Password

Discard Changes

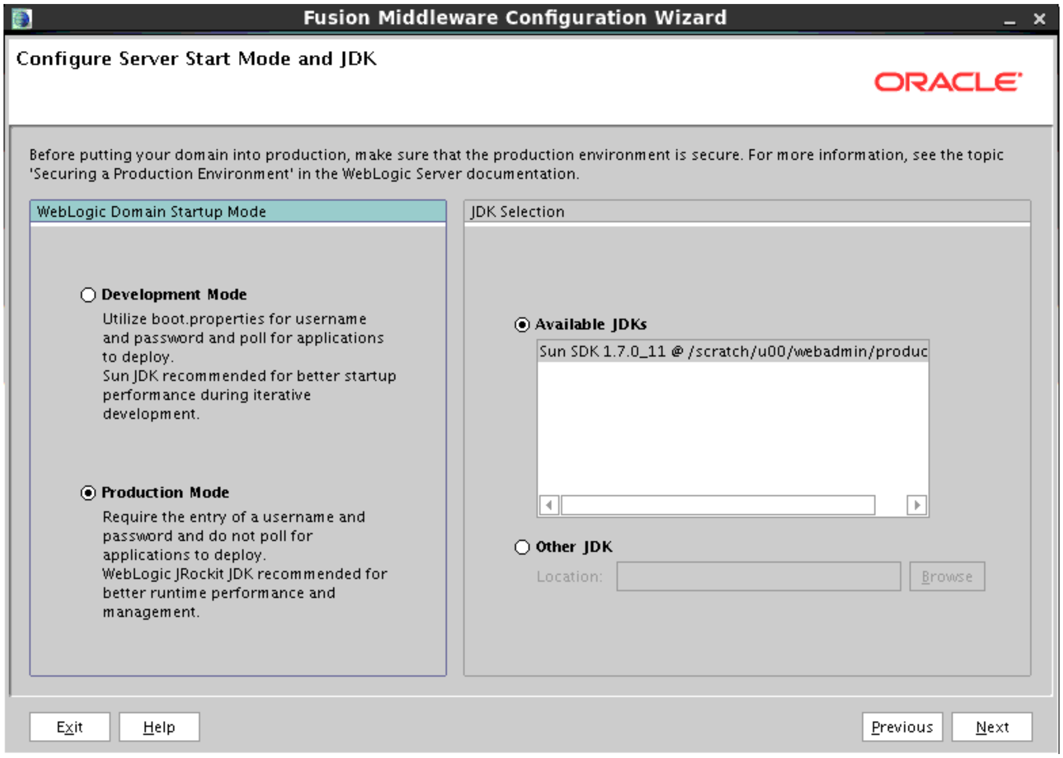
*Name: weblogic

*User password: *********

*Confirm user password: *********

Description: This user is the default administrator.

7. Select **Production Mode** for WebLogic domain Startup Mode and click **Next**.



Fusion Middleware Configuration Wizard

Configure Server Start Mode and JDK

Before putting your domain into production, make sure that the production environment is secure. For more information, see the topic 'Securing a Production Environment' in the WebLogic Server documentation.

WebLogic Domain Startup Mode

○ **Development Mode**
Utilize boot.properties for username and password and poll for applications to deploy.
Sun JDK recommended for better startup performance during iterative development.

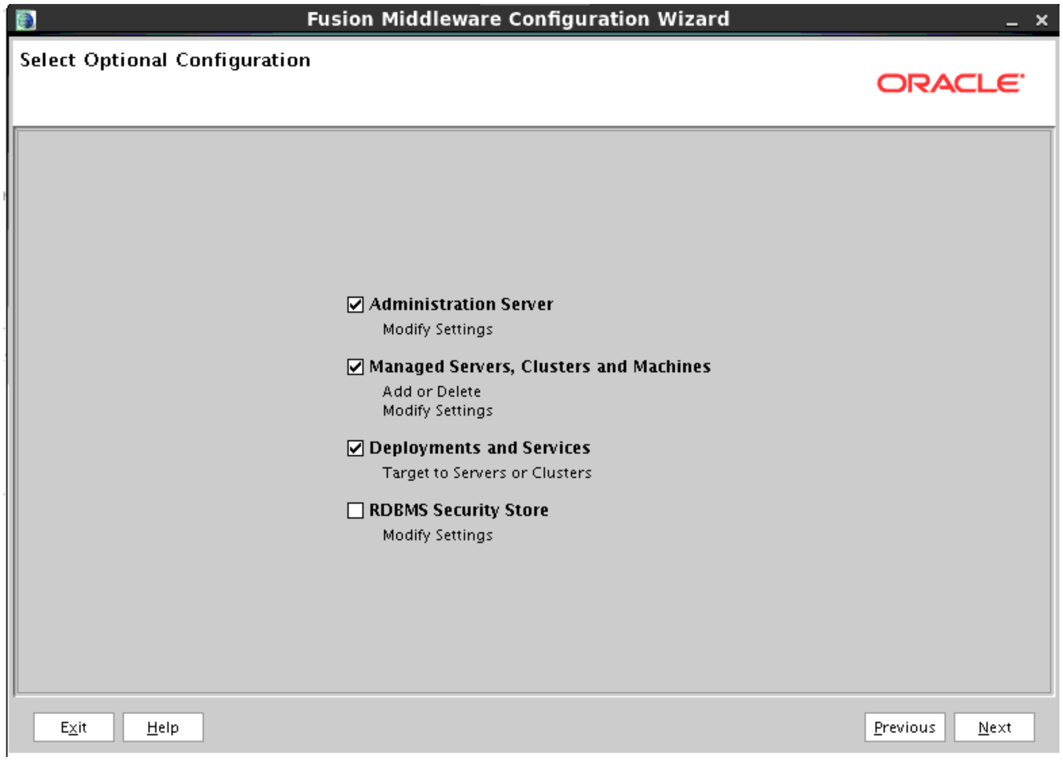
● **Production Mode**
Require the entry of a username and password and do not poll for applications to deploy.
WebLogic JRockit JDK recommended for better runtime performance and management.

JDK Selection

● **Available JDKs**
Sun SDK 1.7.0_11 @ /scratch/u00/webadmin/produc

○ **Other JDK**
Location:

8. Select **Administration Server**, **Managed Servers and Clusters , and Machines** and **Deployments and Services** and click **Next**.


Fusion Middleware Configuration Wizard

Select Optional Configuration

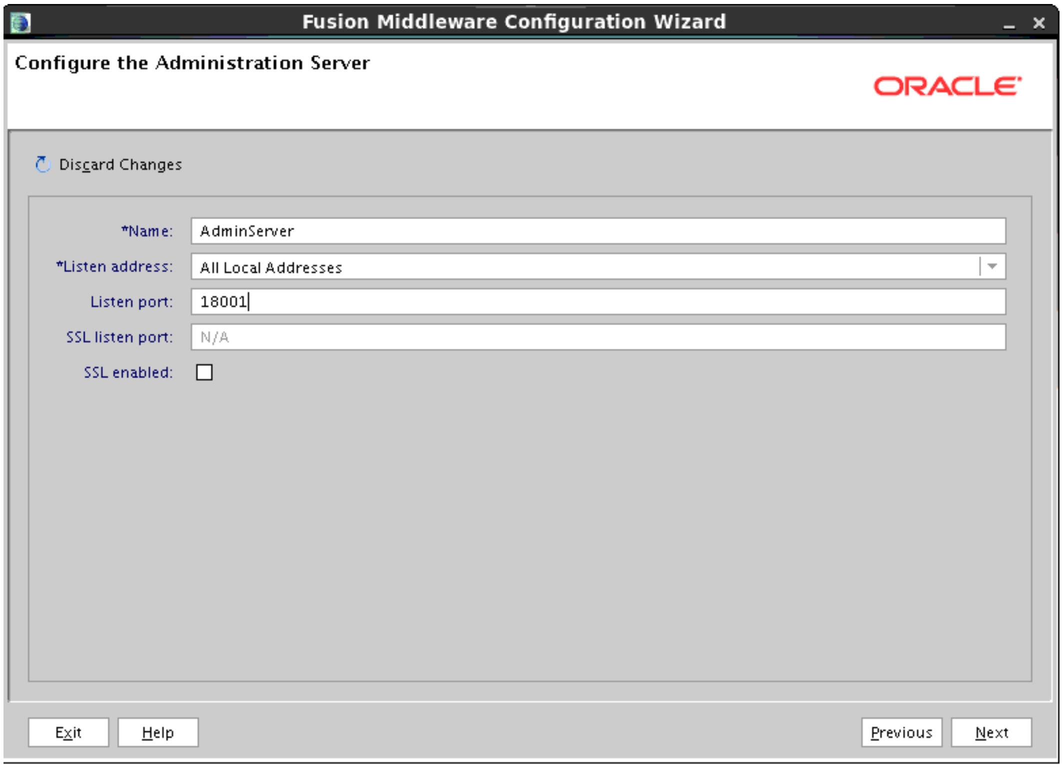
☑ **Administration Server**
Modify Settings

☑ **Managed Servers, Clusters and Machines**
Add or Delete
Modify Settings

☑ **Deployments and Services**
Target to Servers or Clusters

☐ **RDBMS Security Store**
Modify Settings

9. Update the Listen port to one that is not already in use and click **Next**.

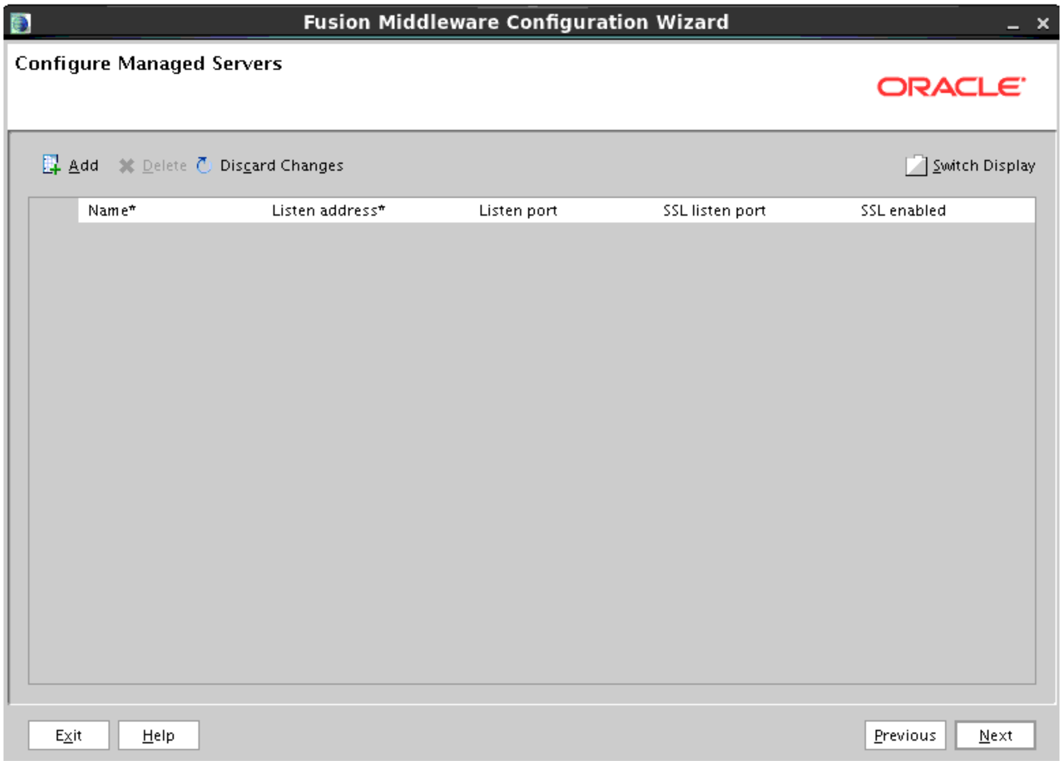

Fusion Middleware Configuration Wizard

Configure the Administration Server

Discard Changes

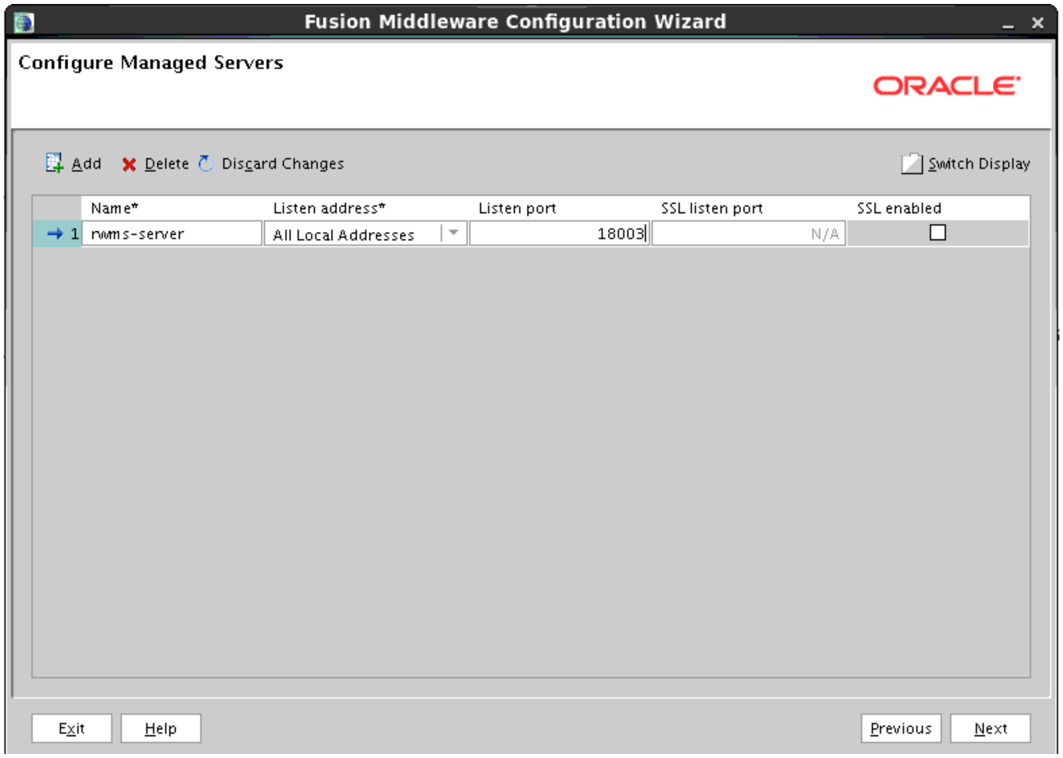
*Name:	AdminServer
*Listen address:	All Local Addresses
Listen port:	18001
SSL listen port:	N/A
SSL enabled:	☐

10. Click the Add button for creating a managed server.



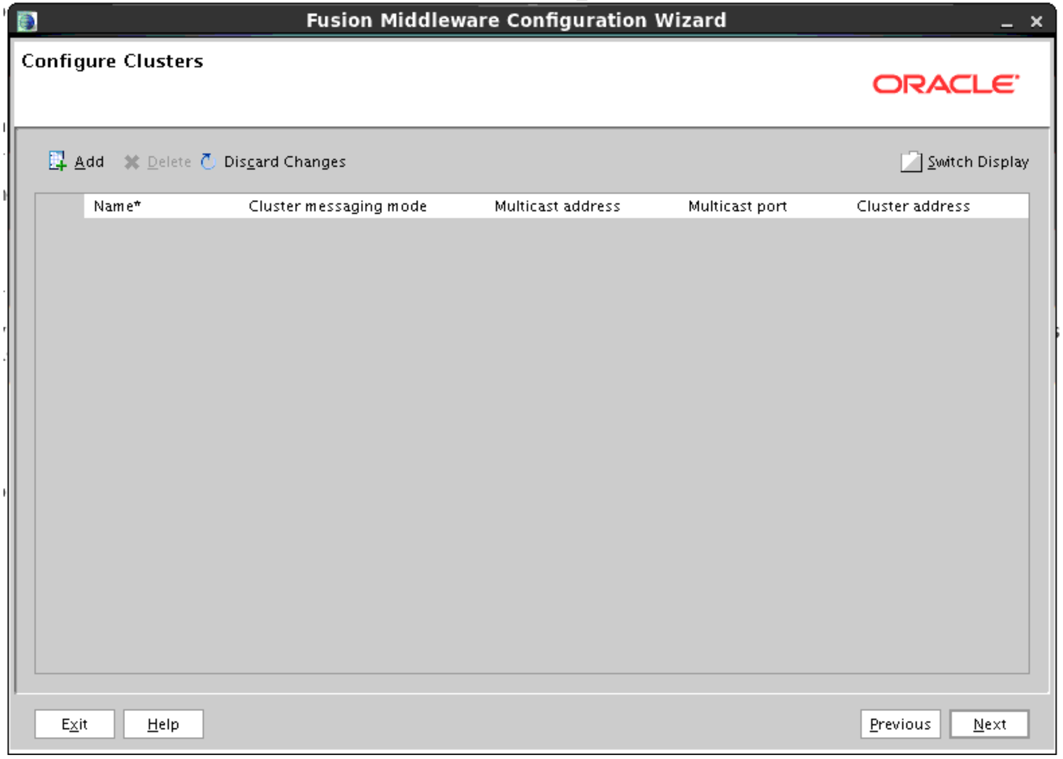
Name*	Listen address*	Listen port	SSL listen port	SSL enabled

11. Enter in the name of the managed server – listen address and listen port. Again be sure to select a port that is not already in use.



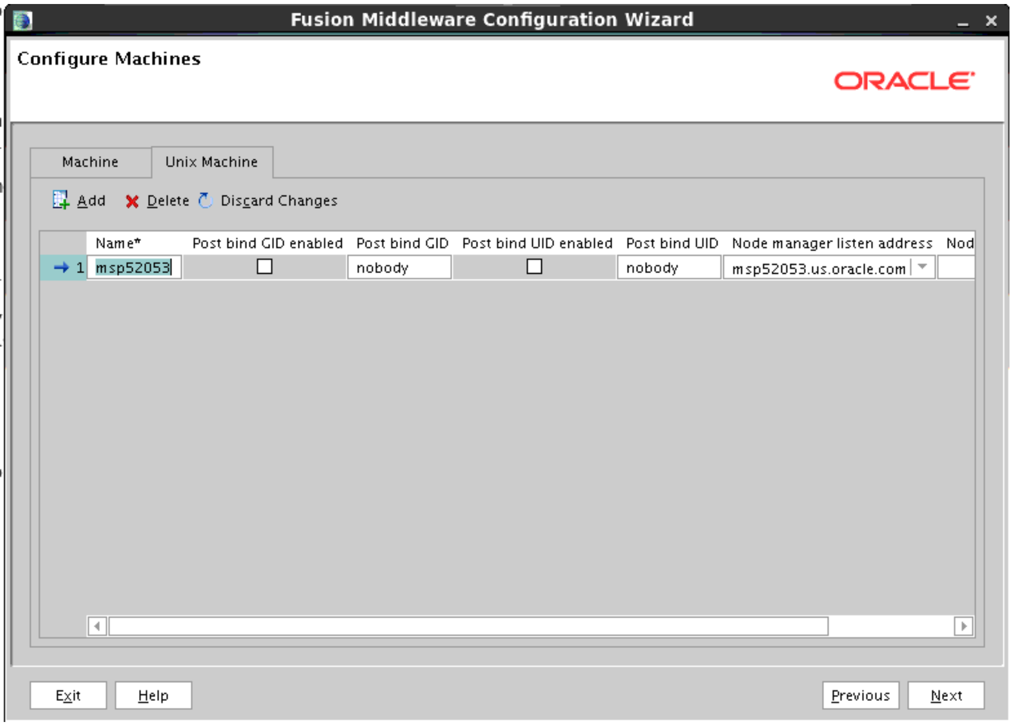
	Name*	Listen address*	Listen port	SSL listen port	SSL enabled
1	rwms-server	All Local Addresses	18003	N/A	□

12. Click **Next** if you are not using a cluster. If you are using a cluster, provide the cluster details and then click **Next**.



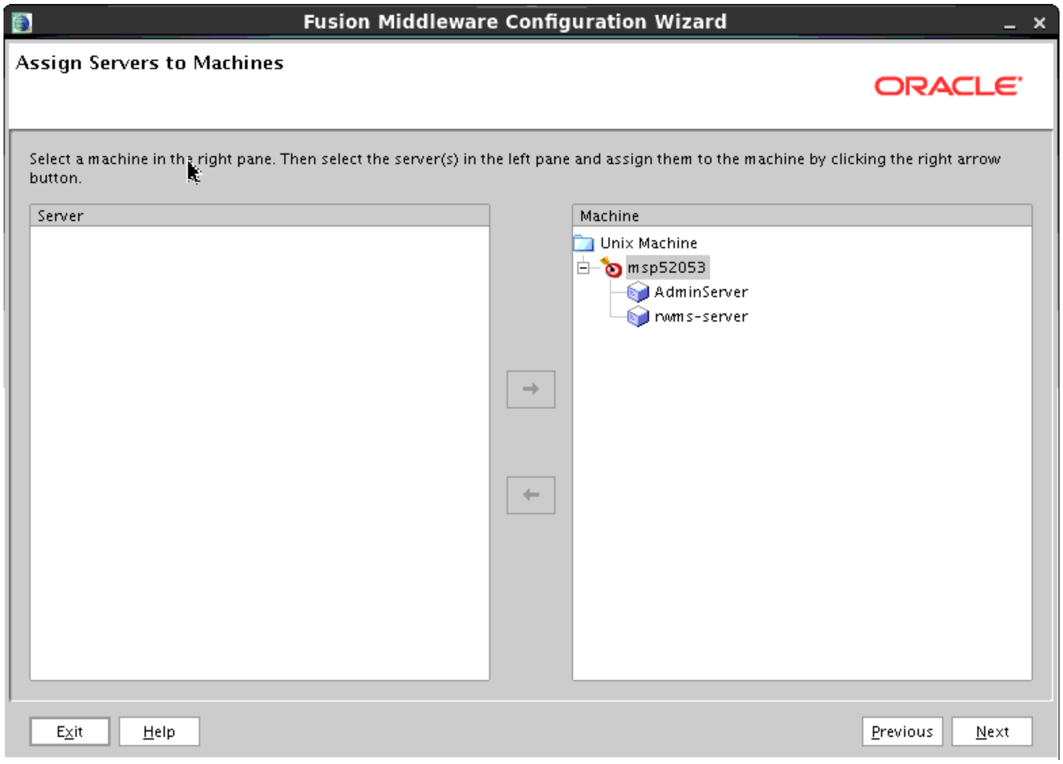
Name*	Cluster messaging mode	Multicast address	Multicast port	Cluster address

13. Add the machine information as a Unix machine. Make sure that the port specified for the Node Manager Listen port is not already in use. Click **Next**.



	Name*	Post bind GID enabled	Post bind GID	Post bind UID enabled	Post bind UID	Node manager listen address	Nod
1	msp52053	☐	nobody	☐	nobody	msp52053.us.oracle.com	

14. Assign the servers to the Unix machine and click **Next**.



Fusion Middleware Configuration Wizard

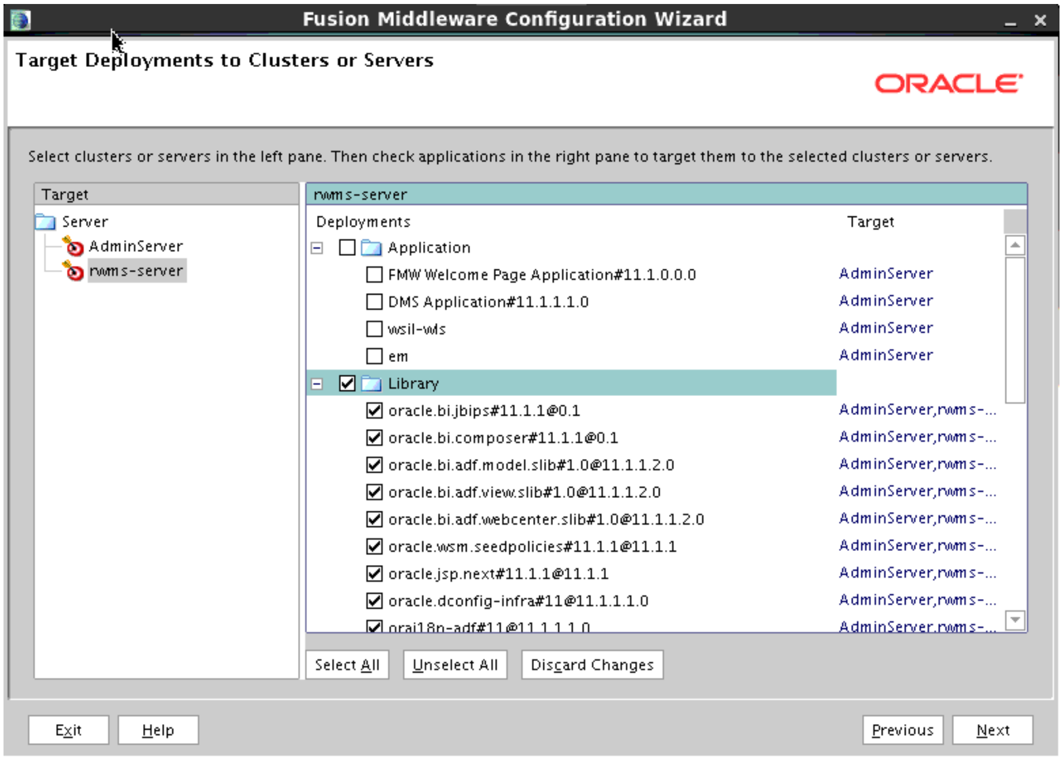
Assign Servers to Machines

Select a machine in the right pane. Then select the server(s) in the left pane and assign them to the machine by clicking the right arrow button.

Server

Machine
- Unix Machine
  - msp52053
    - AdminServer
    - rwms-server

15. Add the ADF libraries to the rwms-server .



Fusion Middleware Configuration Wizard

Target Deployments to Clusters or Servers

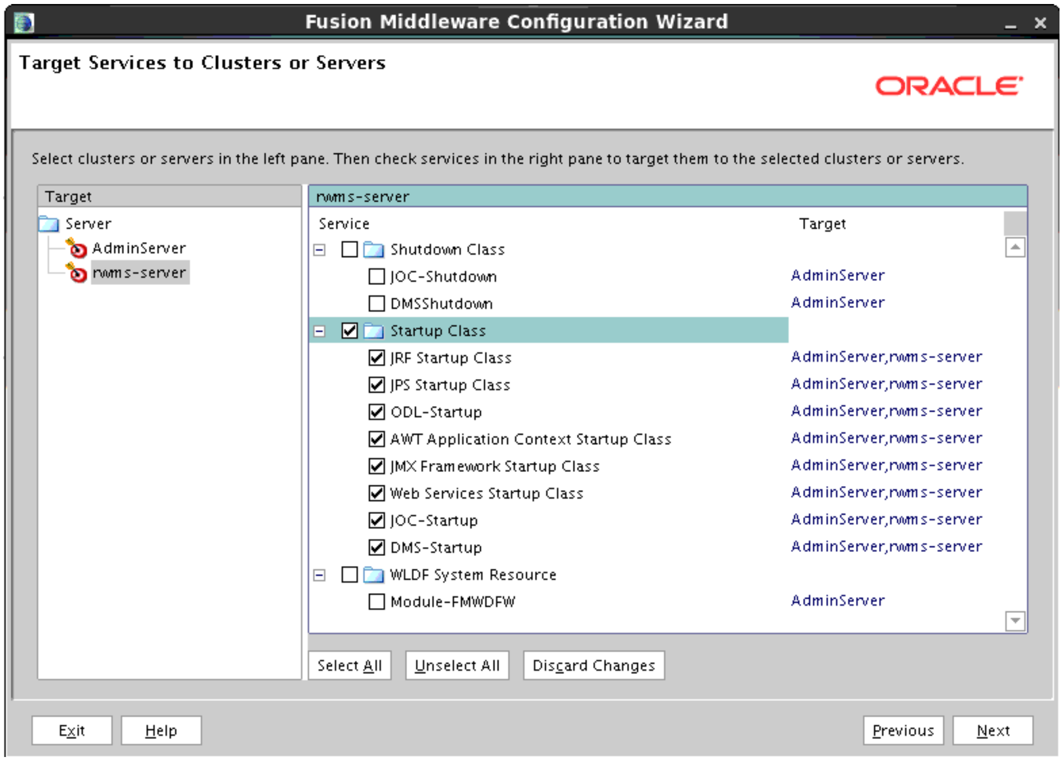
Select clusters or servers in the left pane. Then check applications in the right pane to target them to the selected clusters or servers.

Target
- Server
  - AdminServer
  - rwms-server

rwms-server

Deployments	Target
⊟ □ Application	
□ FMW Welcome Page Application#11.1.0.0.0	AdminServer
□ DMS Application#11.1.1.1.0	AdminServer
□ wsil-wls	AdminServer
□ em	AdminServer
⊟ ☑ Library	
☑ oracle.bi.jbips#11.1.1@0.1	AdminServer,rwms-...
☑ oracle.bi.composer#11.1.1@0.1	AdminServer,rwms-...
☑ oracle.bi.adf.model.slib#1.0@11.1.1.2.0	AdminServer,rwms-...
☑ oracle.bi.adf.view.slib#1.0@11.1.1.2.0	AdminServer,rwms-...
☑ oracle.bi.adf.webcenter.slib#1.0@11.1.1.2.0	AdminServer,rwms-...
☑ oracle.wsm.seedpolicies#11.1.1@11.1.1	AdminServer,rwms-...
☑ oracle.jsp.next#11.1.1@11.1.1	AdminServer,rwms-...
☑ oracle.dconfig-infra#11@11.1.1.1.0	AdminServer,rwms-...
☑ orai18n-adf#11@11.1.1.1.0	AdminServer,rwms-...

16. Add the ADF Startup Classes to the rwms-server.



Target Services to Clusters or Servers

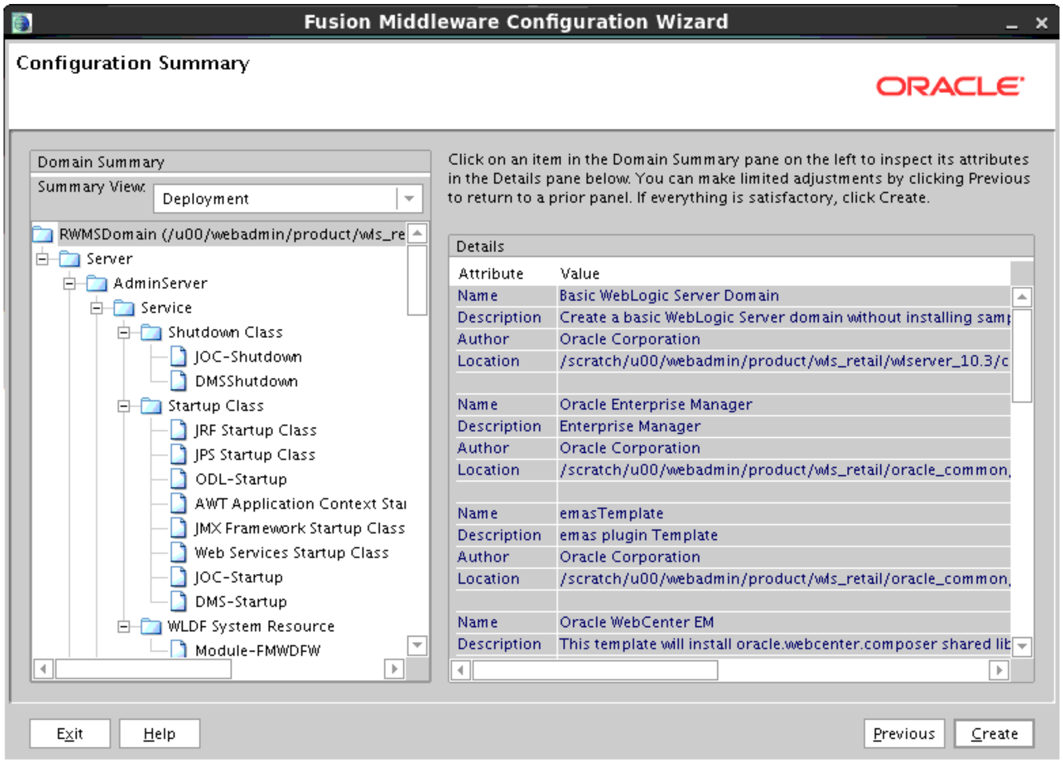
Select clusters or servers in the left pane. Then check services in the right pane to target them to the selected clusters or servers.

Target
- Server
  - AdminServer
  - rwms-server

rwms-server

Service	Target
⊟ □ Shutdown Class	
□ JOC-Shutdown	AdminServer
□ DMSShutdown	AdminServer
⊟ ☑ Startup Class	
☑ JRF Startup Class	AdminServer,rwms-server
☑ JPS Startup Class	AdminServer,rwms-server
☑ ODL-Startup	AdminServer,rwms-server
☑ AWT Application Context Startup Class	AdminServer,rwms-server
☑ JMX Framework Startup Class	AdminServer,rwms-server
☑ Web Services Startup Class	AdminServer,rwms-server
☑ JOC-Startup	AdminServer,rwms-server
☑ DMS-Startup	AdminServer,rwms-server
⊟ □ WLDF System Resource	
□ Module-FMWDFW	AdminServer

4 Click **Create** to create the Domain.



Configuration Summary

Domain Summary

Summary View: Deployment

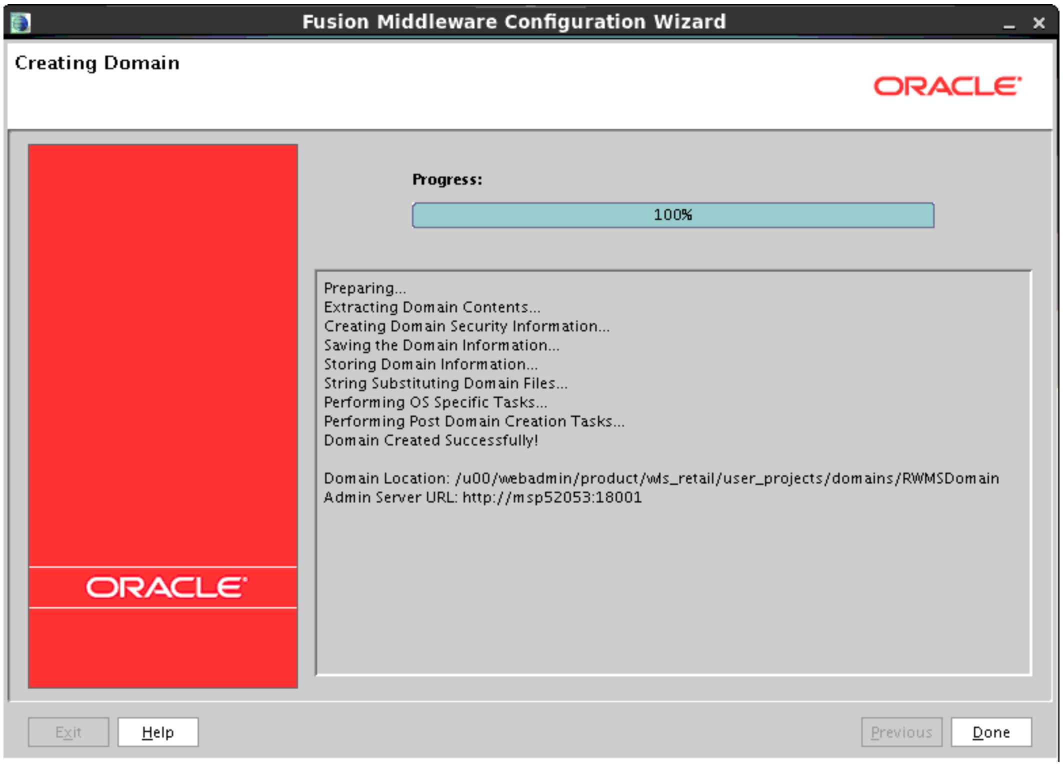
- RWMSDomain (/u00/webadmin/product/wls_re
  - Server
    - AdminServer
      - Service
        - Shutdown Class
          - JOC-Shutdown
          - DMSShutdown
        - Startup Class
          - JRF Startup Class
          - JPS Startup Class
          - ODL-Startup
          - AWT Application Context Stai
          - JMX Framework Startup Class
          - Web Services Startup Class
          - JOC-Startup
          - DMS-Startup
        - WLDF System Resource
          - Module-FMWDFW

Click on an item in the Domain Summary pane on the left to inspect its attributes in the Details pane below. You can make limited adjustments by clicking Previous to return to a prior panel. If everything is satisfactory, click Create.

Details

Attribute	Value
Name	Basic WebLogic Server Domain
Description	Create a basic WebLogic Server domain without installing samp
Author	Oracle Corporation
Location	/scratch/u00/webadmin/product/wls_retail/wlserver_10.3/c
Name	Oracle Enterprise Manager
Description	Enterprise Manager
Author	Oracle Corporation
Location	/scratch/u00/webadmin/product/wls_retail/oracle_common,
Name	emasTemplate
Description	emas plugin Template
Author	Oracle Corporation
Location	/scratch/u00/webadmin/product/wls_retail/oracle_common,
Name	Oracle WebCenter EM
Description	This template will install oracle.webcenter.composer shared lit

17. Click **Done**.



```
Preparing...
Extracting Domain Contents...
Creating Domain Security Information...
Saving the Domain Information...
Storing Domain Information...
String Substituting Domain Files...
Performing OS Specific Tasks...
Performing Post Domain Creation Tasks...
Domain Created Successfully!

Domain Location: /u00/webadmin/product/wls_retail/user_projects/domains/RWMSDomain
Admin Server URL: http://msp52053:18001
```

Start the RWMSDomain and the Node Manager

Start up both the domain and the nodemanager. Edit the nodemanager.properties file at the following location with the below values:

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

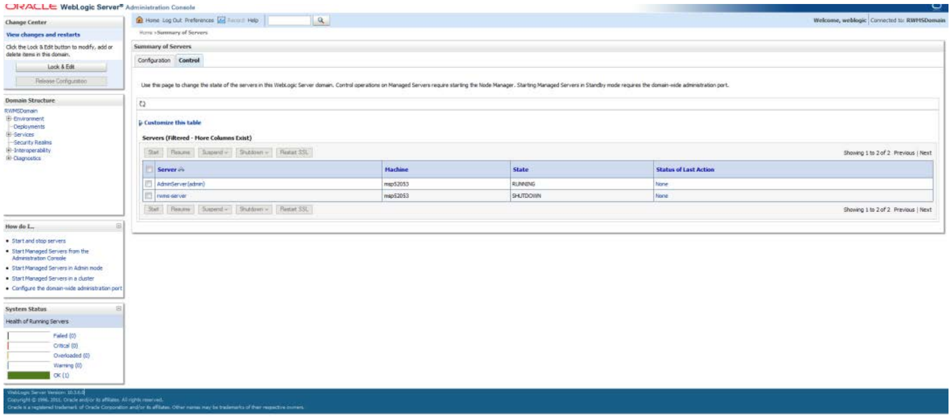
- StartScriptEnabled=true
- StartScriptName=startWebLogic.sh.

After making changes to the nodemanager.properties file, NodeManager must be restarted.

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

Start the Managed Server

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


ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic | Connected to: RWMSDomain
Home >Summary of Servers
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
RWMSDomain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Start and stop servers
Start Managed Servers from the Administration Console
Start Managed Servers in Admin mode
Start Managed Servers in a cluster
Configure the domain-wide administration port
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (1)
Summary of Servers
Configuration
Control
Customize this table
Start
Resume
Suspend
Shutdown
Restart SSL
Showing 1 to 2 of 2 Previous | Next

Click the Lock & Edit button to modify, add or delete items in this domain.

Use this page to change the state of the servers in this WebLogic Server domain. Control operations on Managed Servers require starting the Node Manager. Starting Managed Servers in Standby mode requires the domain-wide administration port.

Servers (Filtered - More Columns Exist)

	Server	Machine	State	Status of Last Action
□	AdminServer(admin)	msp52053	RUNNING	None
□	rwms-server	msp52053	SHUTDOWN	None

WebLogic Server Version: 10.3.6.0
Copyright © 1996, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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
J2EE_ORACLE_HOME	The location of the WebLogic installation that contains the RWMS Enhanced Navigation and Dashboards Domain.	J2EE_ORACLE_HOME=/u00/webadmin/product/wls_retail export J2EE_ORACLE_HOME
J2EE_DOMAIN_HOME	The location of the WebLogic domain that contains the RWMS Enhanced Navigation and Dashboards managed server.	J2EE_DOMAIN_HOME=/u00/webadmin/product/wls_retail/user_projects/domains/RWMSDomain export J2EE_DOMAIN_HOME
DISPLAY	Address and display number of X server on desktop system of user running install. Optional for Enhanced Navigation and Dashboards installation	DISPLAY=<IP address>:0 export DISPLAY
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.AL32UTF8 export NLS_LANG

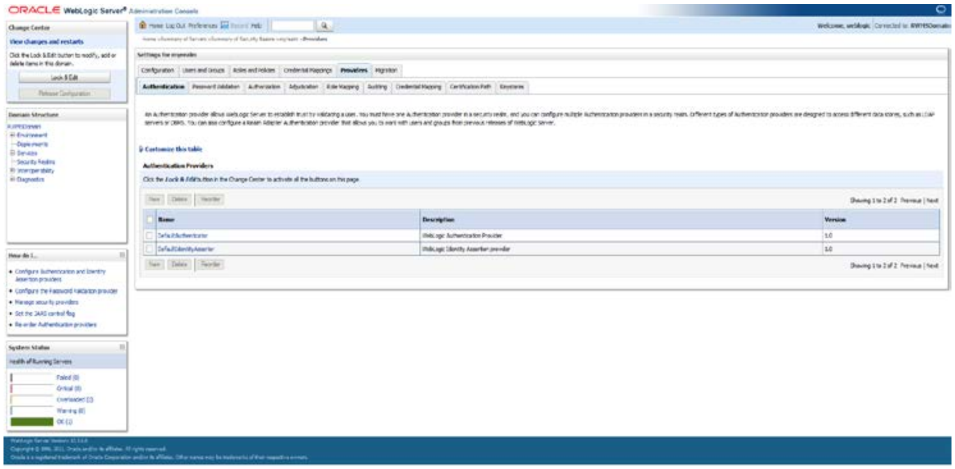
Variable	Description	Example
JAVA_HOME	JAVA_HOME should point to your valid JDK installation.	JAVA_HOME=/vol.rtk/java/os_platform/jdk1.7.0_45.64bit

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. Shut down all the servers of the Weblogic Domain created
2. Once you extract the RWMS installer to < STAGING_DIR > copy the oracle.retail.rwms.security.weblogic.jar present in < STAGING_DIR >/ rwms/installer/rwms14/Enhanced_Navigation_and_Dashboards/Clientlib to the WEBLOGIC_DOMAIN_HOME/lib
3. Start the domain admin server.
4. Log into the WebLogic console.
5. Navigate to: security realms -> myrealm (default realm) -> providers.



Authentication Providers

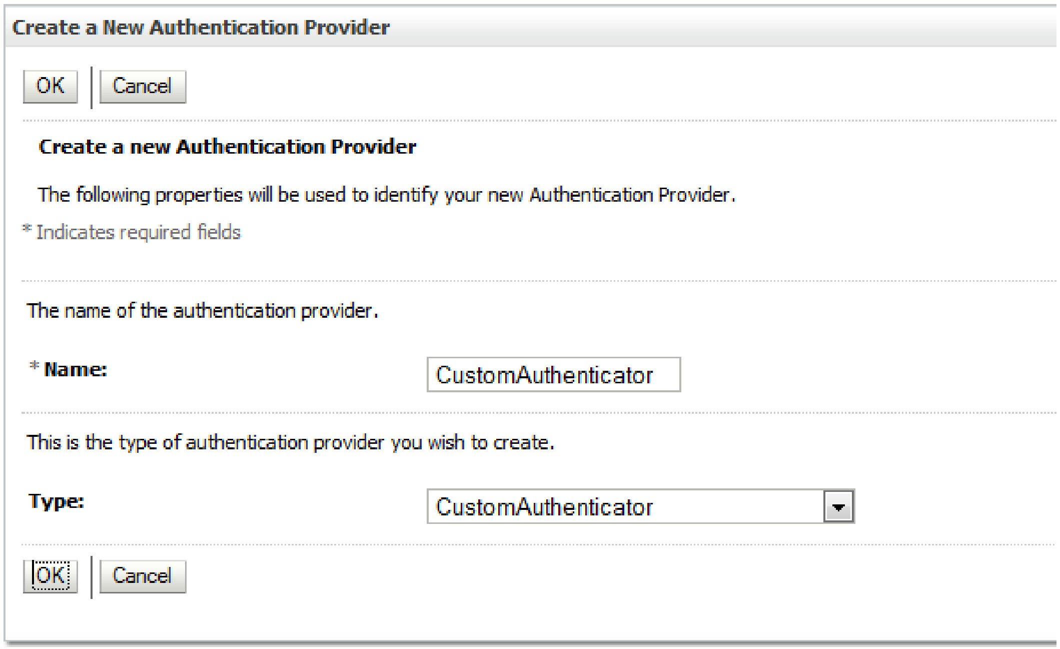
Click the Lock & Edit button in the Change Center to activate all the buttons on this page.

Showing 1 to 2 of 2 Previous | Next

	Name	Description	Version
□	DefaultAuthenticator	WebLogic Authentication Provider	1.0
□	DefaultIdentityAsserter	WebLogic Identity Asserter provider	1.0

6. Start a Lock and Edit session.
7. Click New provider.
8. Select the provider type from the list: Custom Authenticator.

9. Set the provider name (Default: Custom Authenticator).



Create a New Authentication Provider

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.

* Indicates required fields

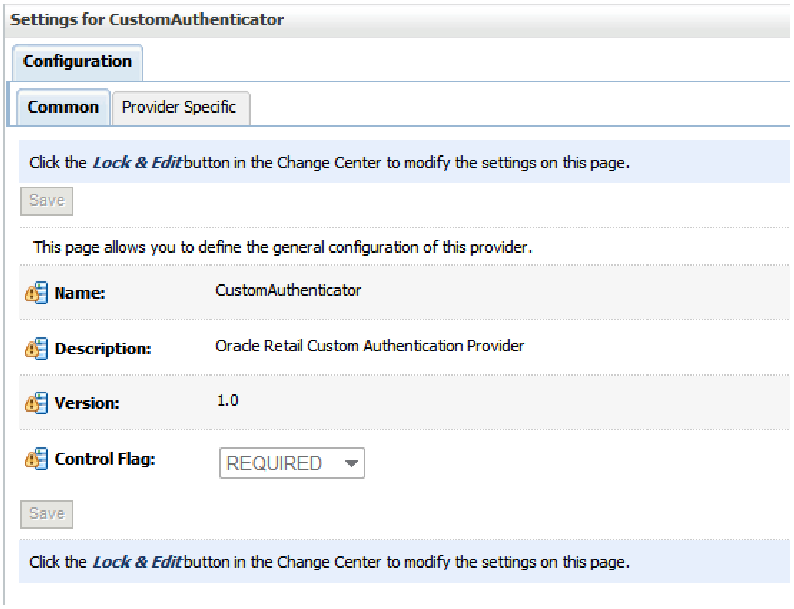
The name of the authentication provider.

* **Name:** CustomAuthenticator

This is the type of authentication provider you wish to create.

Type: CustomAuthenticator

10. Click **Ok**.
11. Open the new provider configuration.
12. Under Common, set the Control Flag to REQUIRED.



Settings for CustomAuthenticator

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

This page allows you to define the general configuration of this provider.

Name: CustomAuthenticator

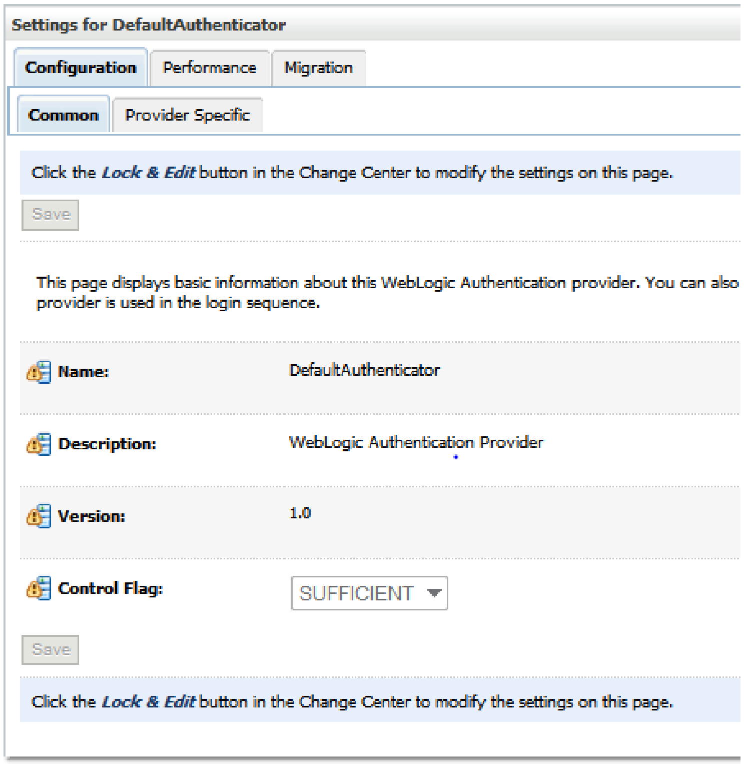
Description: Oracle Retail Custom Authentication Provider

Version: 1.0

Control Flag: REQUIRED

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

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



Settings for DefaultAuthenticator

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

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

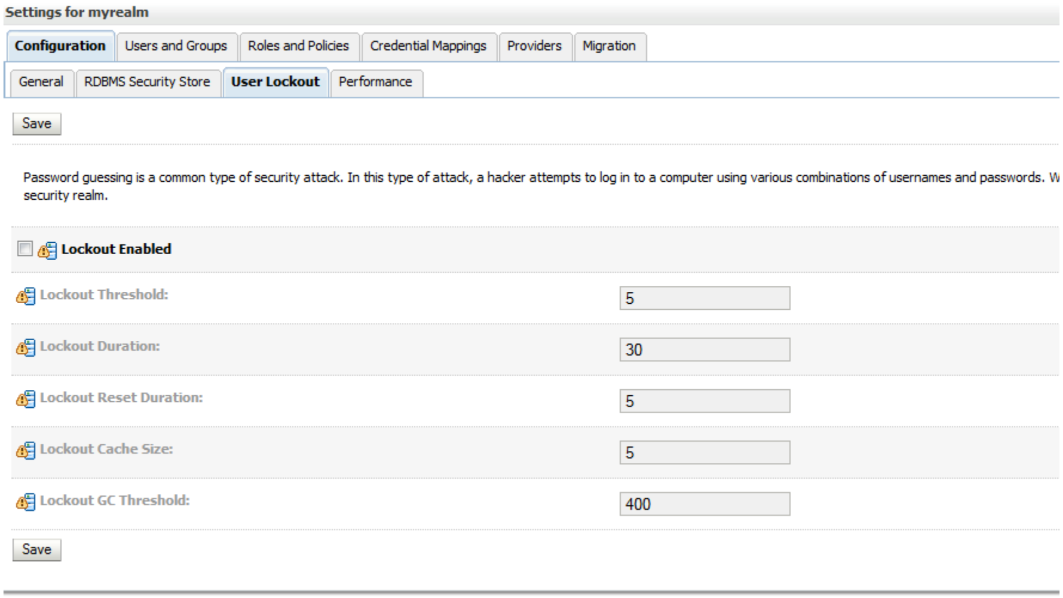
Control Flag: SUFFICIENT

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

14. The order of the providers must be 1st the Default Authenticator, next the Default Identity Asserter and last the Custom Authenticator. Re-order them accordingly.
15. Activate the changes and restart the entire domain.

Disable User Lock Out Settings

1. Login into the Domain.
2. Go to SecurityRealms and then to my relam. Under the configuration tab go to User lock out.
3. Take lock and Edit and disable the Lockout Enabled check box.
4. Activate the changes and Restart the domain.



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

Application: Logging File Settings

For getting clear logs of the application the managed server's logging.xml file should be modified as follows:

Under <log-handlers> tag:

```
<log_handler name='rwmsapp-handler'
class='oracle.core.ojdl.logging.ODLHandlerFactory'
filter='oracle.dfw.incident.IncidentDetectionLogFilter'>
   <property name='path' value='${domain.home}/servers/rwms-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>
```

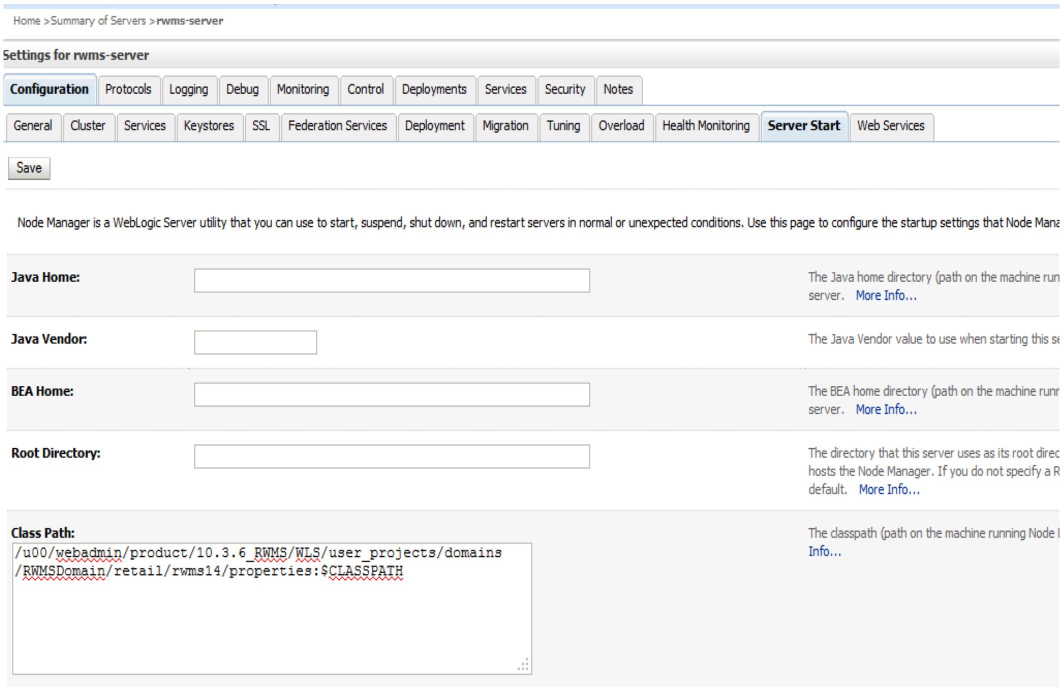
Set the Class Path for the Managed Server

1. Login into the WebLogic console of the RWMS Domain
2. Go to the servers tab, click on rwms-managed server and then under configuration, click on the server start tab
3. Click on lock and Edit and add the bellow content in under classpath

   ```
   <DOMAIN_LOCATION>/retail/<installation_name_you_used
   in_installer>/properties:$CLASSPATH
   ```

   Example:

   ```
   /u00/webadmin/product/10.3.6_RWMS/WLS/user_projects/domains/RWMSDomain/retail/
   rwms14/properties:$CLASSPATH
   ```



Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Mana

Java Home: The Java home directory (path on the machine run server. More Info...

Java Vendor: The Java Vendor value to use when starting this se

BEA Home: The BEA home directory (path on the machine runr server. More Info...

Root Directory: The directory that this server uses as its root direc hosts the Node Manager. If you do not specify a R default. More Info...

Class Path:

```
/u00/webadmin/product/10.3.6_RWMS/WLS/user_projects/domains
/RWMSDomain/retail/rwms14/properties:$CLASSPATH
```

The classpath (path on the machine running Node I Info...

4. Click on Save and Activate the changes.
5. Restart the rwms managed servers so that the changes will be picked by the server.

Manual Installation and Configuration Tasks

This section covers the installation and configuration of external components and other dependencies for RWMS 14.0.

Prerequisites

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

- The RWMS Database Schema installation into a running Oracle 11gR2 database platform must be completed.
- The RWMS application components must be installed into a working WebLogic 10.3.6 domain.
- The RWMS Enhanced Navigation and Dashboard must be installed into a working WebLogic 10.3.6 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 14.0 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	rwms14inst
<WLS_JAVA_HOME>	The JDK location used by the WebLogic Server instance being installed to.	/u00/webadmin/java/jdk1.7.0_45.64bit
<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/bipublisher/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/ClassicDo main/config/fmwconfig/servers/ WLS_FORMS
<FORMS_INSTALL_BASE>	The Oracle Forms base configuration directory.	<WLS_SERVER_CONFIG_DIR>/ applications/ formsapp_11.1.2 /config
<RWMS_INSTALL_DIR>	The final installation directory for RWMS application to run from (such as forms, extras, and reports).	/u00/webadmin/rwms14inst/ base/

Note: In the following sections, the Oracle BI EE 11g 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 10.3.6. 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 "software only 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 11.2.0.3 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. Configure the BI Publisher repository. Set security model, add users, assign roles, add reports, add printers, set repository path, set data source, etc.
7. Set up and copy the RWMS BI Publisher Report Templates produced for RWMS.
8. Set up for the RWMS application specific configuration files to integrate BI Publisher with the RWMS online app.

Install Oracle BI EE 11g

1. Run the Repository Creation Utility to create the BiPublisher-related database schemas and other database objects. Create the BIPlatform schema into an existing ORACLE 11.2.0.3 DB

   Note: Download Repository Creation Utility software from http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/downloads/bi-downloads-1525270.html. Install it on your desktop

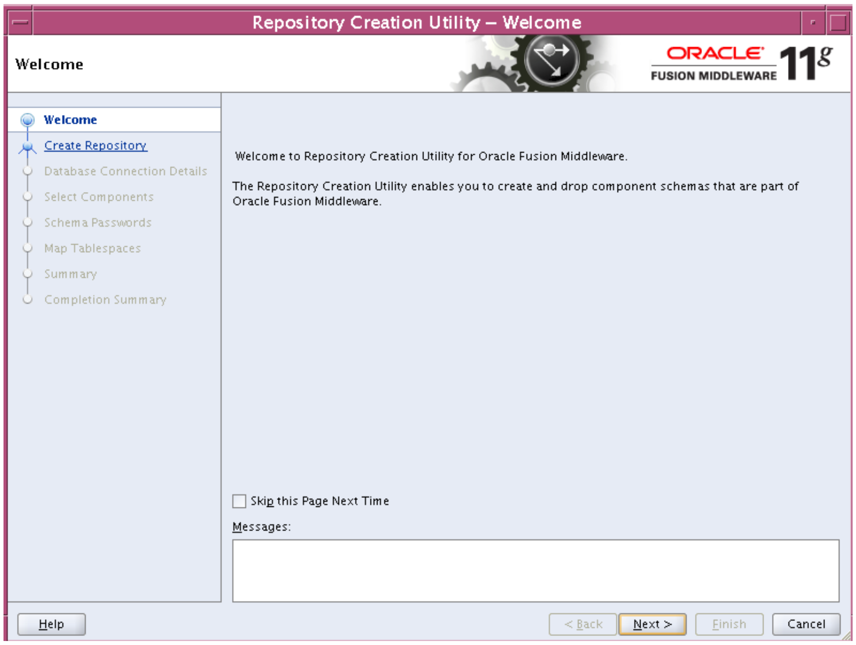
2. Export your DISPLAY

   Example: export DISPLAY=10.141.10.110:0.

3. Go to $RCU_HOME/bin.

   Example: /linux/x86_64/ofm_11g/RCU_11.1.1.6/rcuHome/bin>

   `Start RCU:  ./rcu`



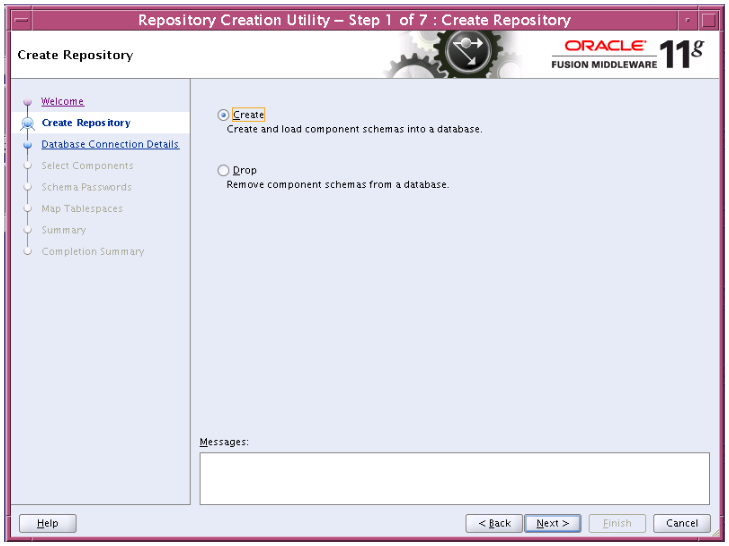
Welcome to Repository Creation Utility for Oracle Fusion Middleware.

The Repository Creation Utility enables you to create and drop component schemas that are part of Oracle Fusion Middleware.

□ Skip this Page Next Time

Messages:

4. Click **Next**

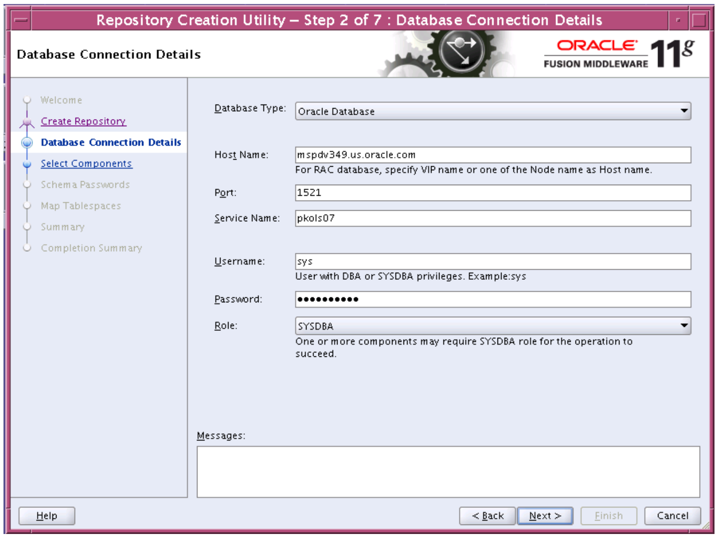


● Create
Create and load component schemas into a database.

○ Drop
Remove component schemas from a database.

Messages:

5. Launch Oracle BI EE RCU Repository Creation Utility to create the Oracle BI EE schemas need for the Oracle BI EE BiPublisher installation. On this screen select Create Repository.



Repository Creation Utility – Step 2 of 7 : Database Connection Details

Database Connection Details

Database Type: Oracle Database

Host Name: mspdv349.us.oracle.com

For RAC database, specify VIP name or one of the Node name as Host name.

Port: 1521

Service Name: pkols07

Username: sys

User with DBA or SYSDBA privileges. Example:sys

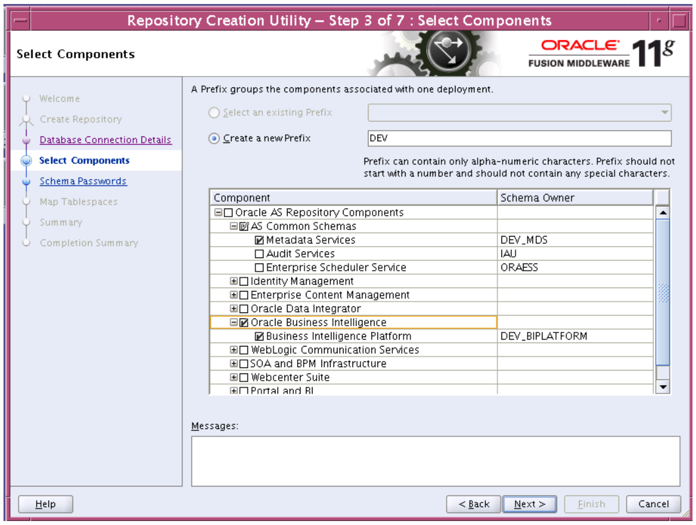
Password: ●●●●●●●●●●

Role: SYSDBA

One or more components may require SYSDBA role for the operation to succeed.

Messages:

6. On the Database Connection Details screen, enter your Oracle Database information.



Repository Creation Utility – Step 3 of 7 : Select Components

Select Components

A Prefix groups the components associated with one deployment.

○ Select an existing Prefix

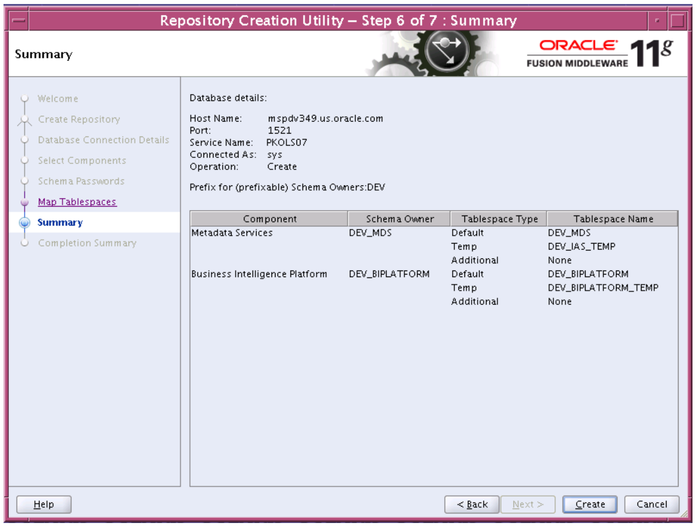
● Create a new Prefix: DEV

Prefix can contain only alpha-numeric characters. Prefix should not start with a number and should not contain any special characters.

Component	Schema Owner
⊟ □ Oracle AS Repository Components	
⊟ ☑ AS Common Schemas	
☑ Metadata Services	DEV_MDS
□ Audit Services	IAU
□ Enterprise Scheduler Service	ORAESS
⊞ □ Identity Management	
⊞ □ Enterprise Content Management	
⊞ □ Oracle Data Integrator	
⊟ ☑ Oracle Business Intelligence	
☑ Business Intelligence Platform	DEV_BIPLATFORM
⊞ □ WebLogic Communication Services	
⊞ □ SOA and BPM Infrastructure	
⊞ □ Webcenter Suite	
⊞ □ Portal and BI	

Messages:

7. On the Select Components screen, select Oracle Business Intelligence check box.



Database details:

Host Name: mspdv349.us.oracle.com
Port: 1521
Service Name: PKOLS07
Connected As: sys
Operation: Create

Prefix for (prefixable) Schema Owners:DEV

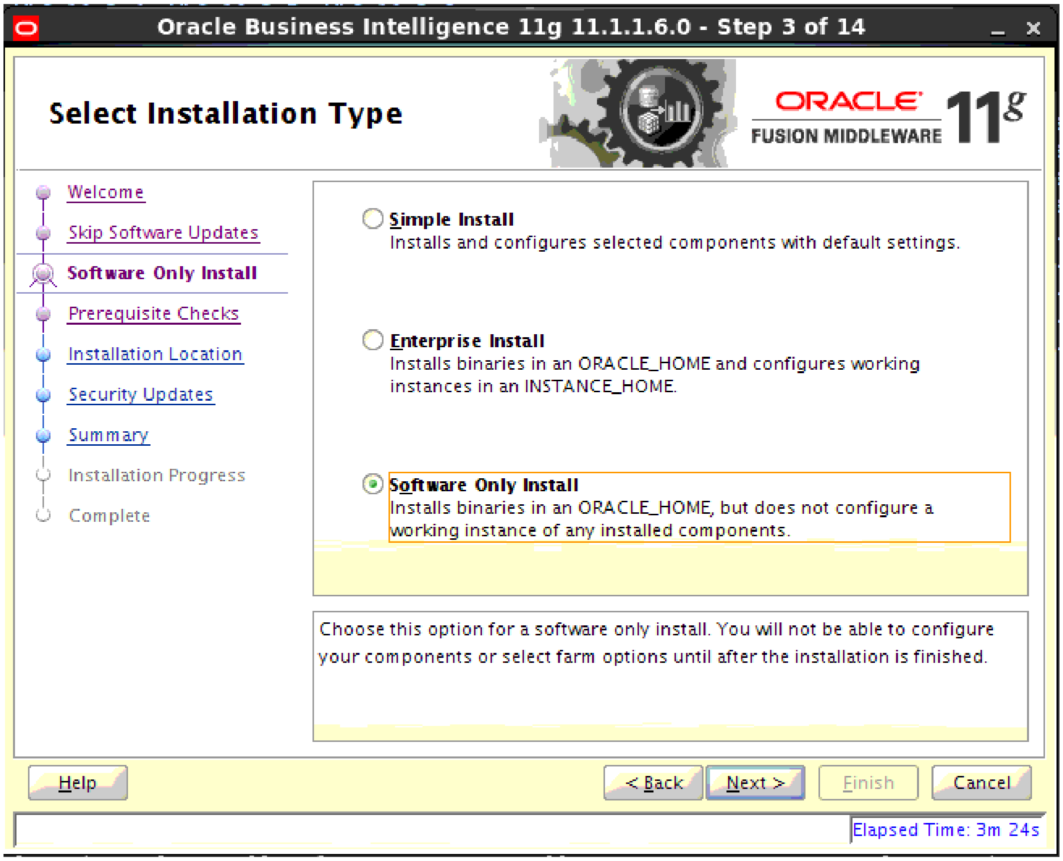
Component	Schema Owner	Tablespace Type	Tablespace Name
Metadata Services	DEV_MDS	Default	DEV_MDS
		Temp	DEV_IAS_TEMP
		Additional	None
Business Intelligence Platform	DEV_BIPLATFORM	Default	DEV_BIPLATFORM
		Temp	DEV_BIPLATFORM_TEMP
		Additional	None

The Summary of the Components created by the RCU tool is displayed.

8. Install a new instance of WebLogic Server 10.3.6 or use an existing one. Having one WebLogic Server for Oracle BI EE-BiPublisher 11g related items is recommended.

9. Install Oracle BI EE and select “Software Only Install”. You launch Oracle BI EE by going to OBIEE_INSTALL/obiee11.1.1.6/bishiphome/Disk1 and entering:

```
./runInstaller
```



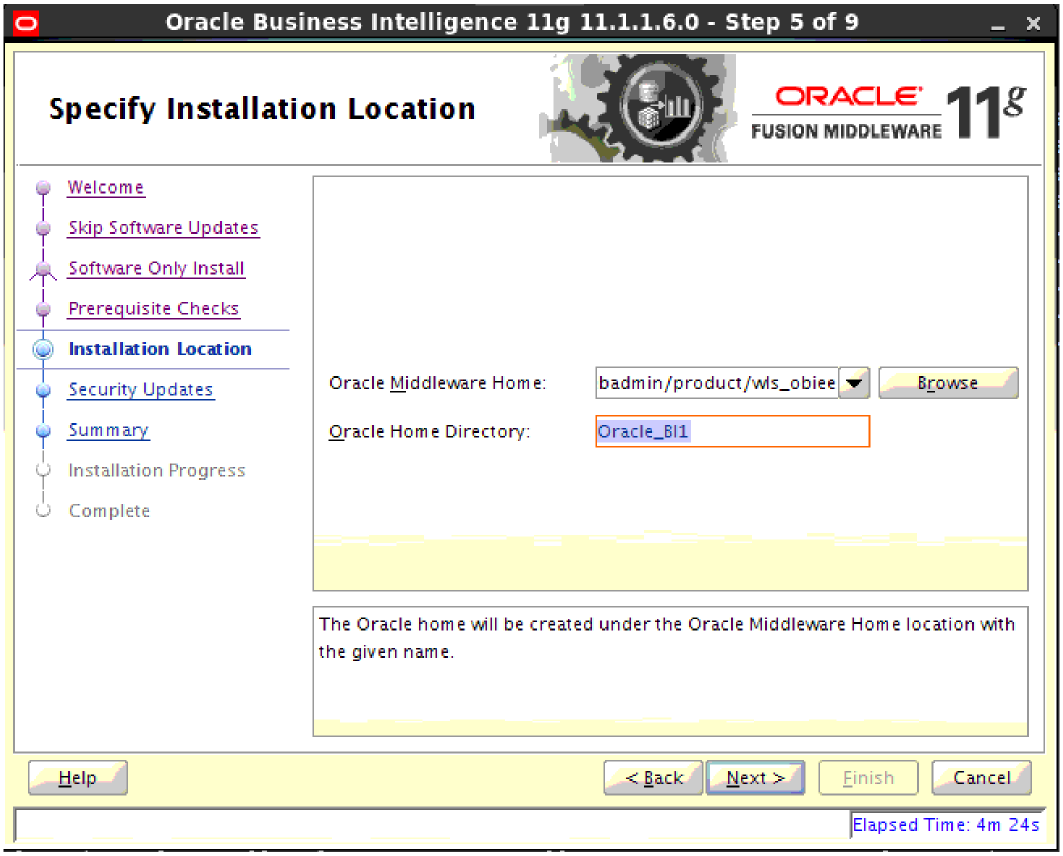
○ Simple Install
Installs and configures selected components with default settings.

○ Enterprise Install
Installs binaries in an ORACLE_HOME and configures working instances in an INSTANCE_HOME.

● Software Only Install
Installs binaries in an ORACLE_HOME, but does not configure a working instance of any installed components.

Choose this option for a software only install. You will not be able to configure your components or select farm options until after the installation is finished.

10. Select the WebLogic home where you want to install Oracle BI EE.



Oracle Middleware Home: badmin/product/wls_obiee

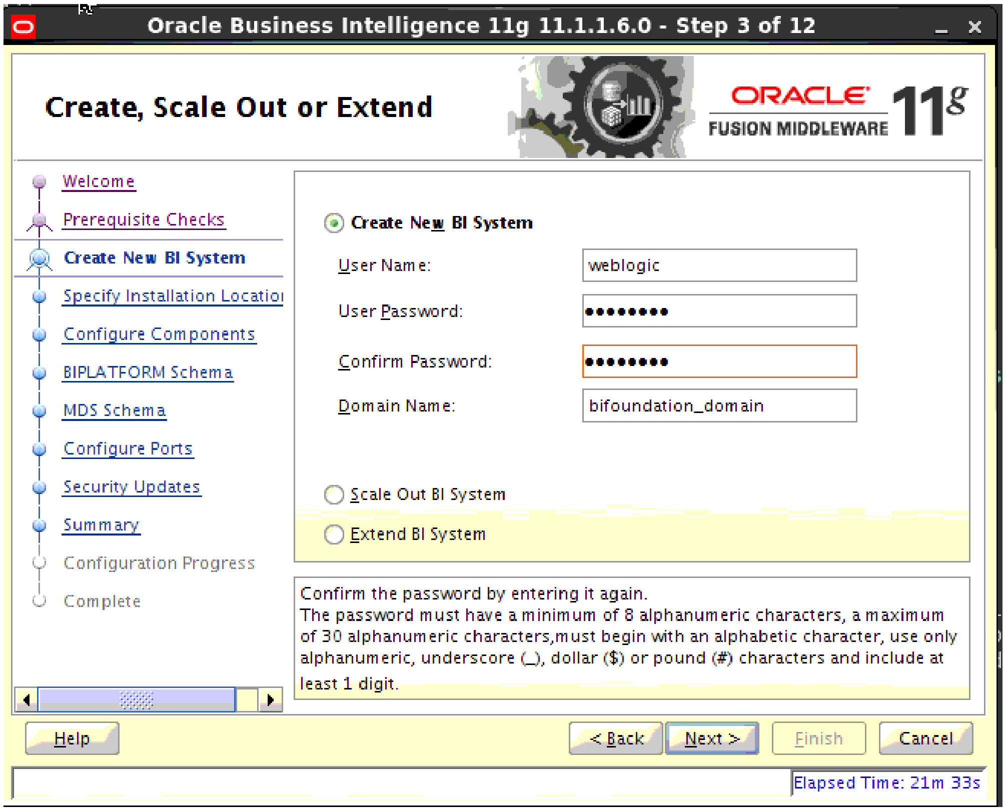
Oracle Home Directory: Oracle_BI1

The Oracle home will be created under the Oracle Middleware Home location with the given name.

Elapsed Time: 4m 24s

11. Proceed to next and finish up the software install. Then go the <WebLogic home>/Oracle_BI1/bin and run config.sh to configure the bipublisher.

12. On the Create or Scale Out BI System screen, you are asked for the WebLogic password and provided with a recommended a Domain Name. Enter and confirm your WebLogic password and accept the recommended Domain Name; "bifoundation domain"



● Create New BI System

User Name: weblogic

User Password: ••••••••

Confirm Password: ••••••••

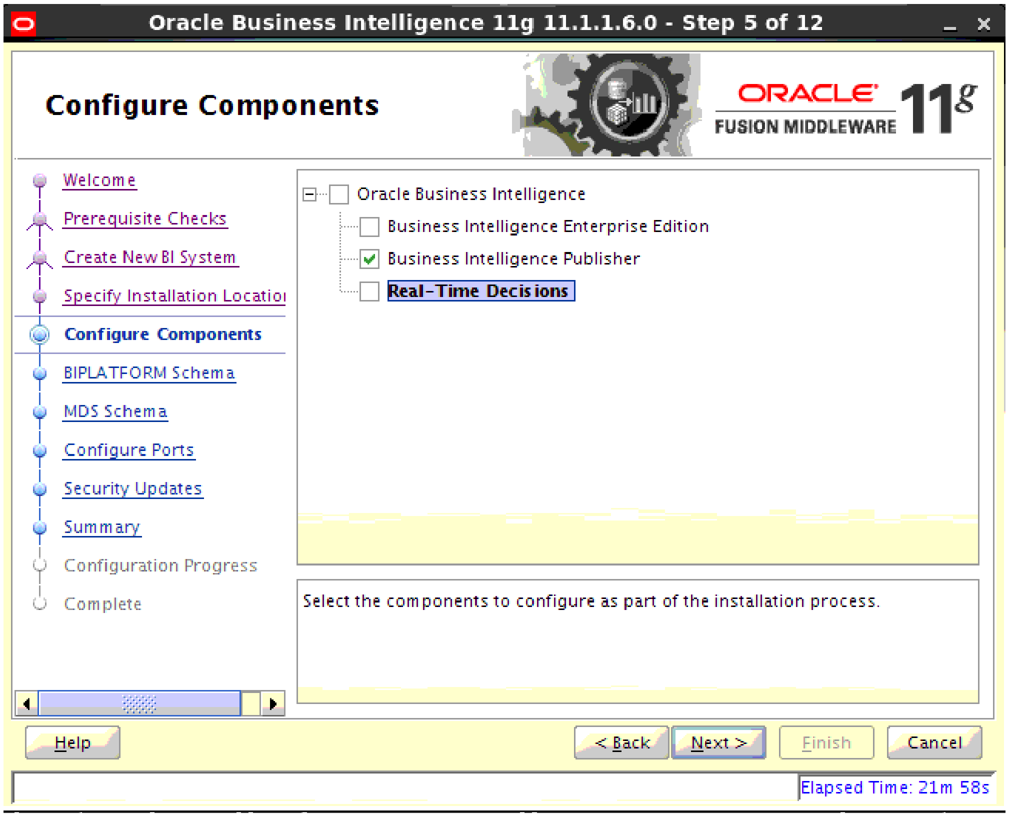
Domain Name: bifoundation_domain

○ Scale Out BI System

○ Extend BI System

Confirm the password by entering it again.
The password must have a minimum of 8 alphanumeric characters, a maximum of 30 alphanumeric characters,must begin with an alphabetic character, use only alphanumeric, underscore (_), dollar ($) or pound (#) characters and include at least 1 digit.

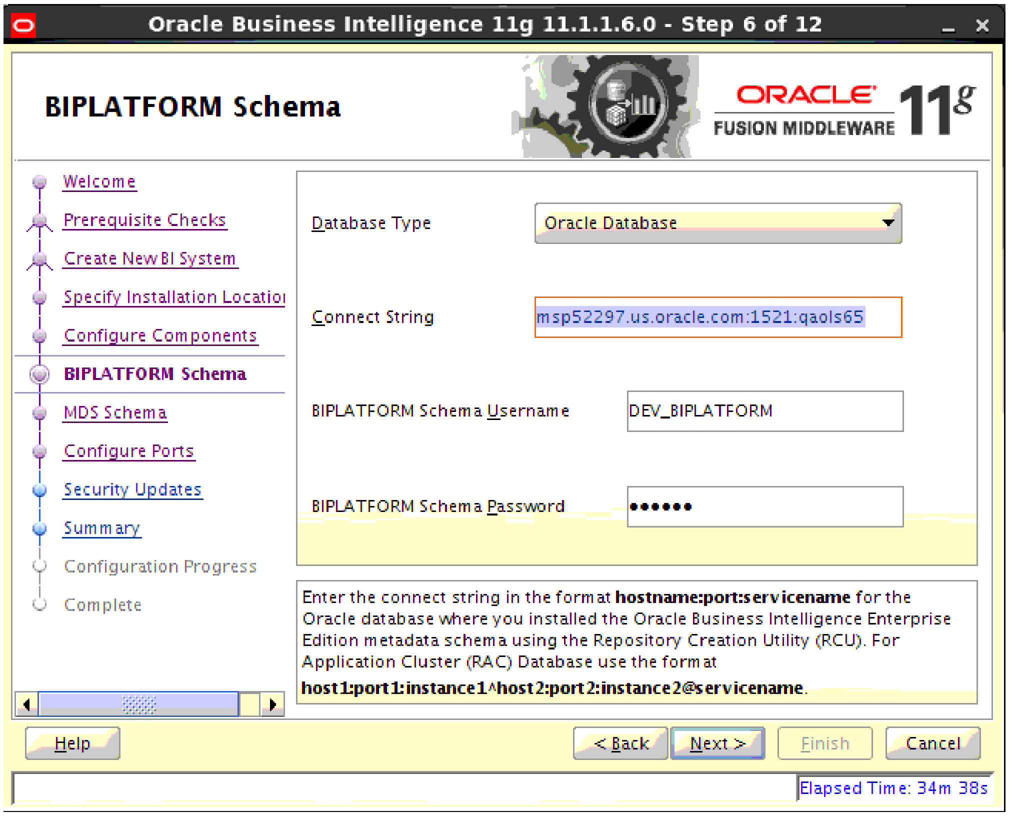
13. Configure Oracle BI EE, create default bifoundation_domain and configure component Business Intelligence Publisher only.



- □ Oracle Business Intelligence
  - □ Business Intelligence Enterprise Edition
  - ☑ Business Intelligence Publisher
  - □ Real-Time Decisions

Select the components to configure as part of the installation process.

14. Enter the data base credentials of the BIPLATFORM schema.



Database Type: Oracle Database

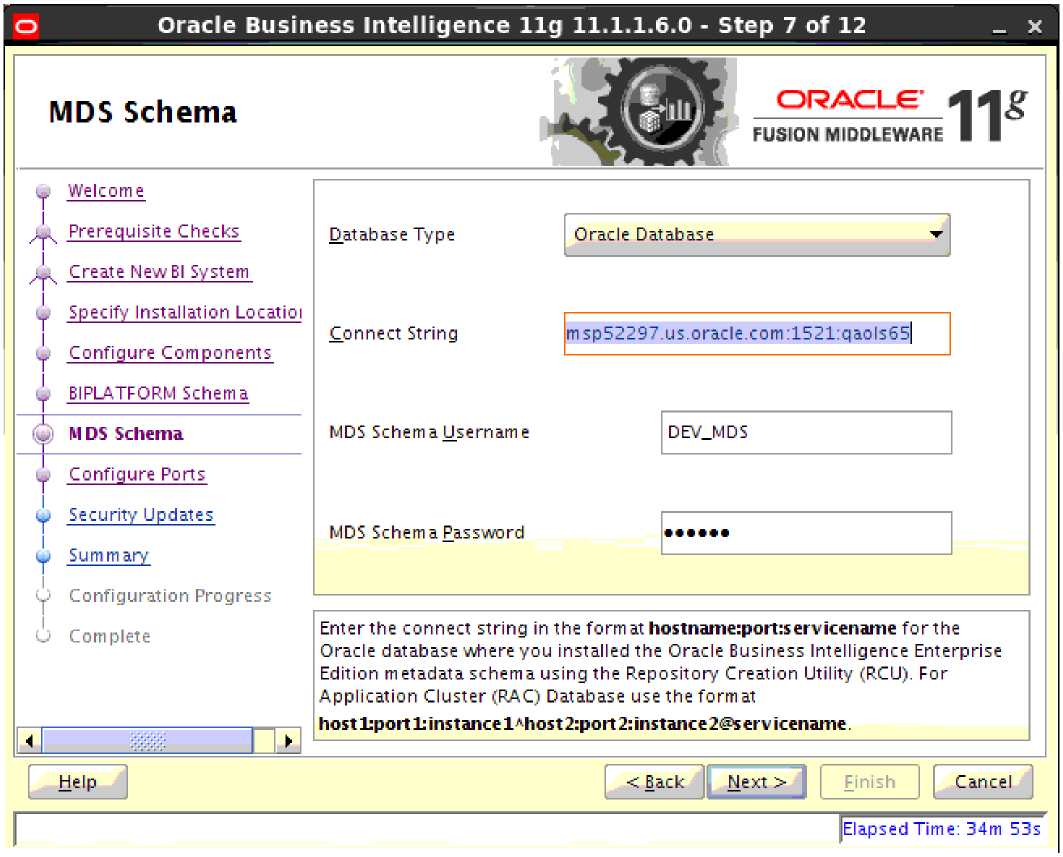
Connect String: msp52297.us.oracle.com:1521:qaols65

BIPLATFORM Schema Username: DEV_BIPLATFORM

BIPLATFORM Schema Password: ••••••

Enter the connect string in the format **hostname:port:servicename** for the Oracle database where you installed the Oracle Business Intelligence Enterprise Edition metadata schema using the Repository Creation Utility (RCU). For Application Cluster (RAC) Database use the format **host1:port1:instance1^host2:port2:instance2@servicename**.

15. Enter the Data base credentials for the MDS Schema.



Database Type: Oracle Database

Connect String: msp52297.us.oracle.com:1521:qaols65

MDS Schema Username: DEV_MDS

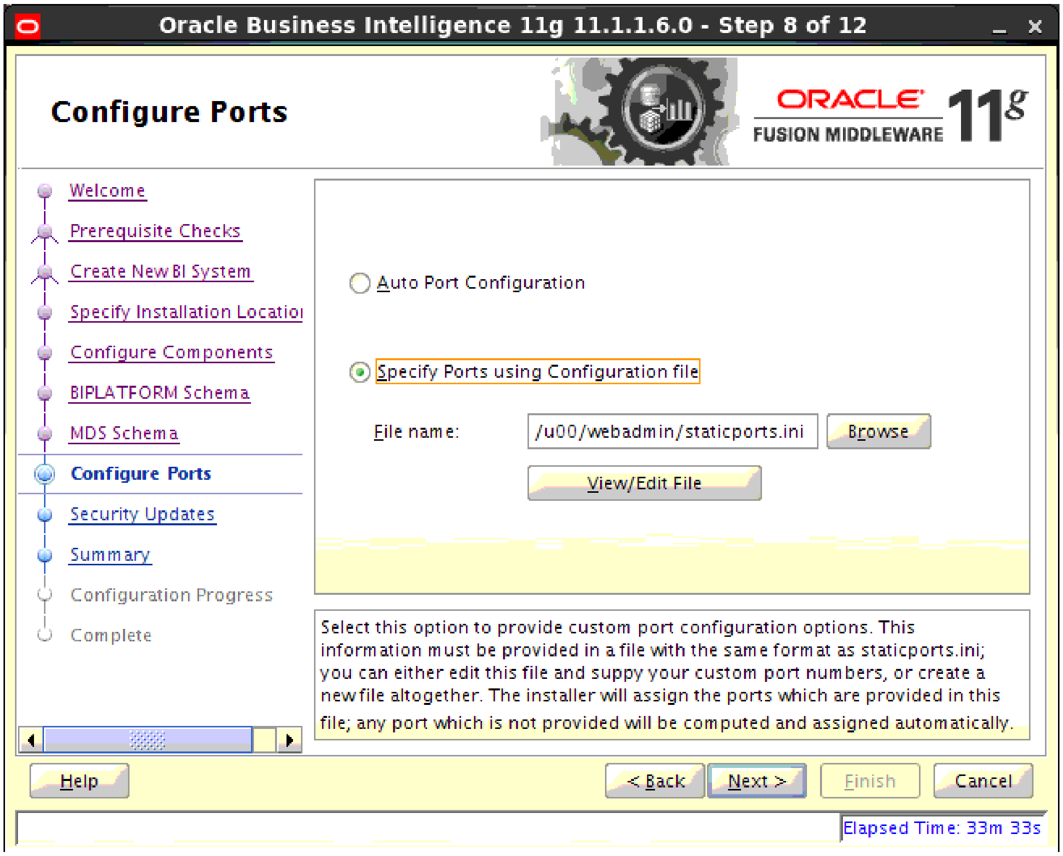
MDS Schema Password: ●●●●●●

Enter the connect string in the format **hostname:port:servicename** for the Oracle database where you installed the Oracle Business Intelligence Enterprise Edition metadata schema using the Repository Creation Utility (RCU). For Application Cluster (RAC) Database use the format **host1:port1:instance1^host2:port2:instance2@servicename**.

Elapsed Time: 34m 53s

16. Configure your BI ports. This screen allows you to assign Oracle BI EE ports from the staticports.ini file.

    This file is located in the Oracle BI EE software at: /obiee11.1.1.6/bishiphome/Disk1/stage/Response/staticports.ini.



Oracle Business Intelligence 11g 11.1.1.6.0 - Step 8 of 12

Configure Ports

○ Auto Port Configuration

● Specify Ports using Configuration file

File name: /u00/webadmin/staticports.ini

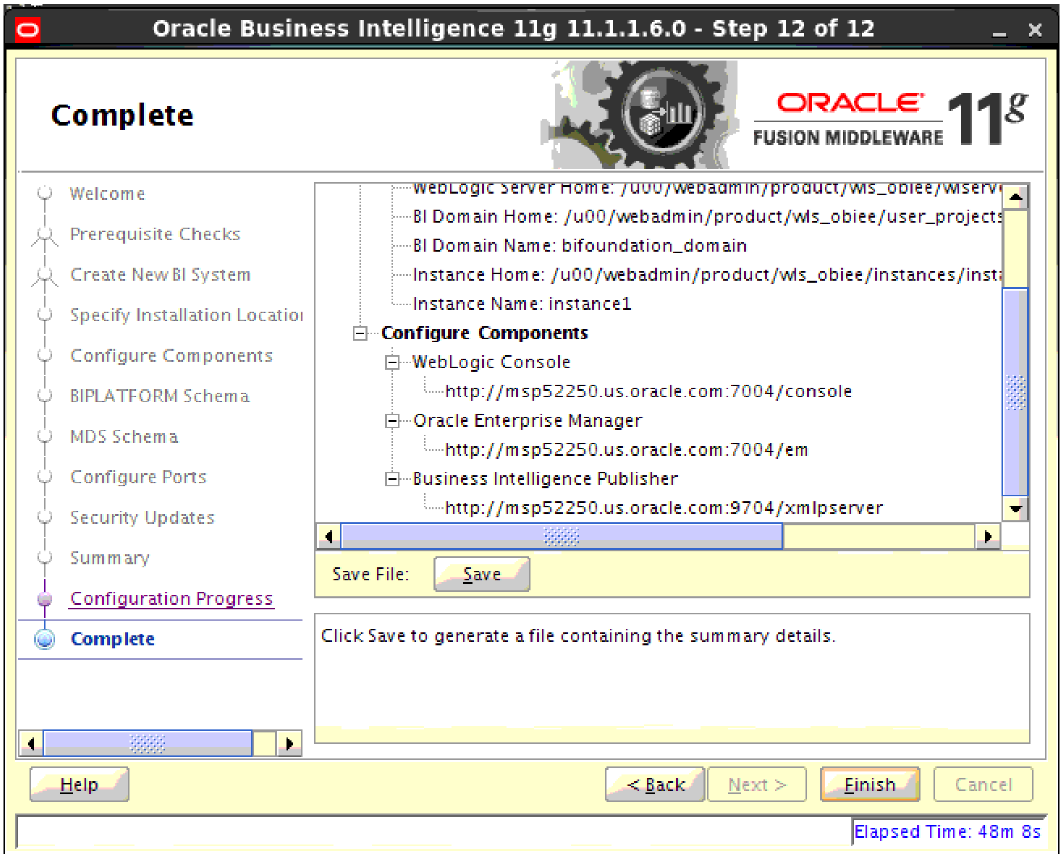
Select this option to provide custom port configuration options. This information must be provided in a file with the same format as staticports.ini; you can either edit this file and suppy your custom port numbers, or create a new file altogether. The installer will assign the ports which are provided in this file; any port which is not provided will be computed and assigned automatically.

Elapsed Time: 33m 33s

17. Edit this file to make sure you will have the ports you want for your BiPublisher components. Otherwise the installer will assign default port numbers.

18. Document and test the URLs that are created.

This screen contains the URL's for the components that got installed.

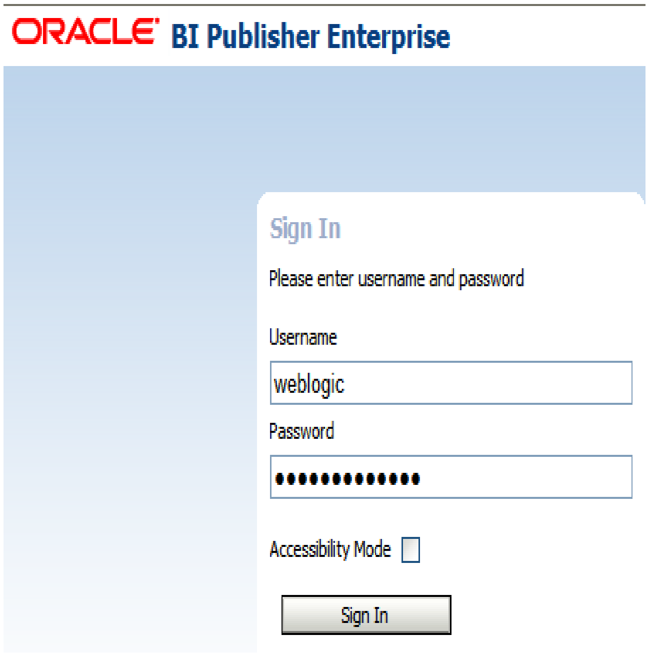


WebLogic Server Home: /u00/webadmin/product/wls_obiee/wlserv
BI Domain Home: /u00/webadmin/product/wls_obiee/user_projects
BI Domain Name: bifoundation_domain
Instance Home: /u00/webadmin/product/wls_obiee/instances/insta
Instance Name: instance1
Configure Components
WebLogic Console
http://msp52250.us.oracle.com:7004/console
Oracle Enterprise Manager
http://msp52250.us.oracle.com:7004/em
Business Intelligence Publisher
http://msp52250.us.oracle.com:9704/xmlpserver

Click Save to generate a file containing the summary details.

Elapsed Time: 48m 8s

19. To test your BIPublisher installation, launch xmlpserver. Login with the credentials you entered in your Oracle BI EE configuration (weblogic / password).



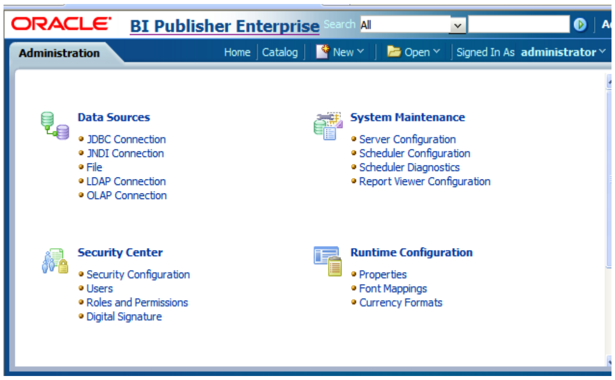
Please enter username and password

Username: weblogic

Password: ●●●●●●●●●●●●●

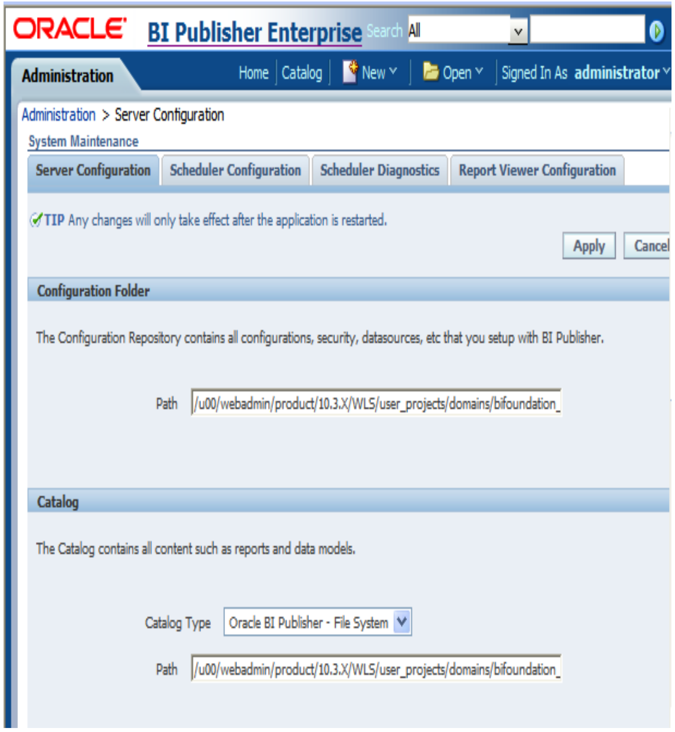
Accessibility Mode □

20. Post install steps: Configure the BI Publisher repository. After signon, select "Administration".



21. On the System Maintenance Section, press Server Configuration

22. Navigate to the Configuration Screen.



TIP Any changes will only take effect after the application is restarted.

Configuration Folder

The Configuration Repository contains all configurations, security, datasources, etc that you setup with BI Publisher.

Path /u00/webadmin/product/10.3.X/WLS/user_projects/domains/bifoundation_

Catalog

The Catalog contains all content such as reports and data models.

Catalog Type Oracle BI Publisher - File System

Path /u00/webadmin/product/10.3.X/WLS/user_projects/domains/bifoundation_

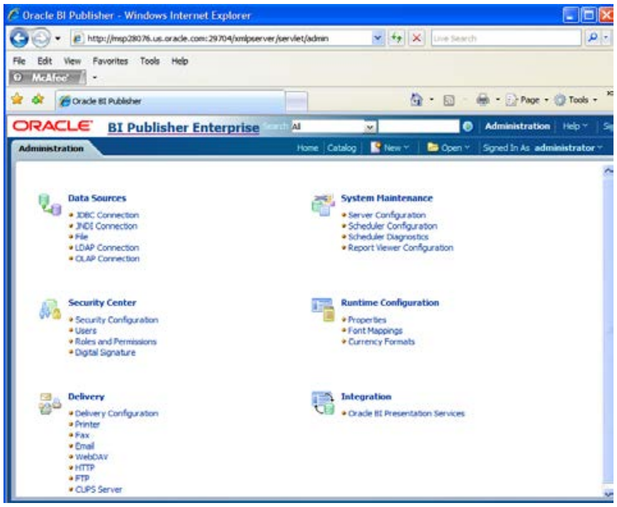
23. On this screen on the Configuration Folder section, enter the path to your repository. On the Catalog section enter Catalog Type: Oracle BI Publisher – File System from the drop down menu.

24. This is the path you entered in the Configuration Section and Catalog Section:

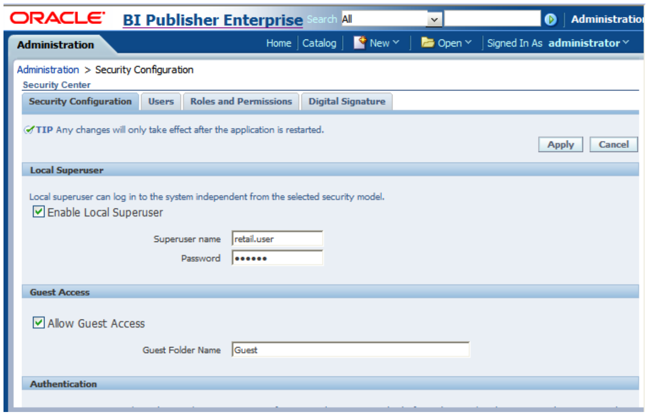
25. $MIDDLEWARE_HOME/user_projects/domains/bifoundation_domain/config/bipublisher/repository

26. Restart the BI Publisher after this change.

27. Post install step: Set BiPublisher security model.



a. On the BiPublisher 11g Administration Screen, click Security Configuration from the Security Center.



TIP Any changes will only take effect after the application is restarted.

Local superuser can log in to the system independent from the selected security model.

☑ Enable Local Superuser

Superuser name: retail.user

Password: ••••••

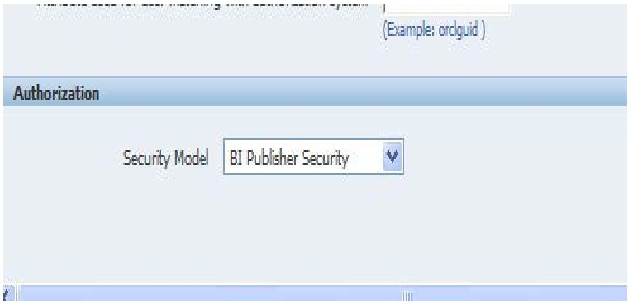
☑ Allow Guest Access

Guest Folder Name: Guest

b. Enable a superuser by checking the “Enable Local SuperUser” box and by entering name and password on the corresponding fields on this screen.

c. Mark “Allow Guest Access” check box. Enter “Guest” as Guest Folder Name.

d. Scroll down the screen and locate the Authorization section:



Security Model: BI Publisher Security

e. Select BI Publisher Security from the Security Model list.

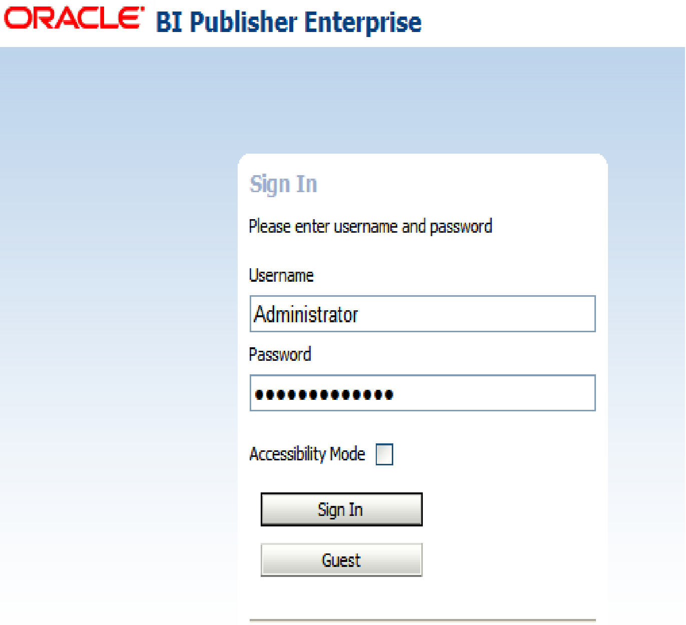
f. The default user name for the BI Publisher Security Model is Administrator.

g. On the password text field, enter a value that you can remember. It is going to be the password for Login to xmlpserver.

h. Save the changes and re-start the BIPublisher server.

i. Launch xmlpserver. To Login you must use the new credentials that you set up in the former step: Username: Administrator Password: password.

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



28. Post install step: Set the repository path.

Example:
/u00/webadmin/product/10.3.X/WLS/user_projects/domains/bifoundation_domain/config/bipublisher/repository In the Oracle BI EE file system you will find the repository in the following location:

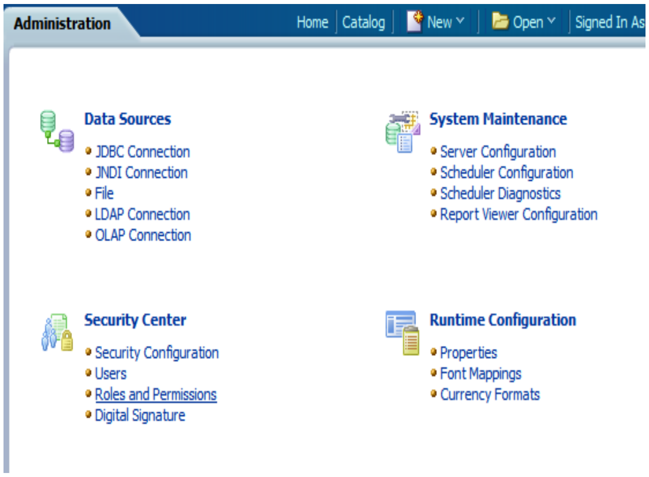
```
$OBIEE/wls/user_projects/domains/bifoundation_domain/config/bipublisher/repository
```

In the repository you will see the following directories:

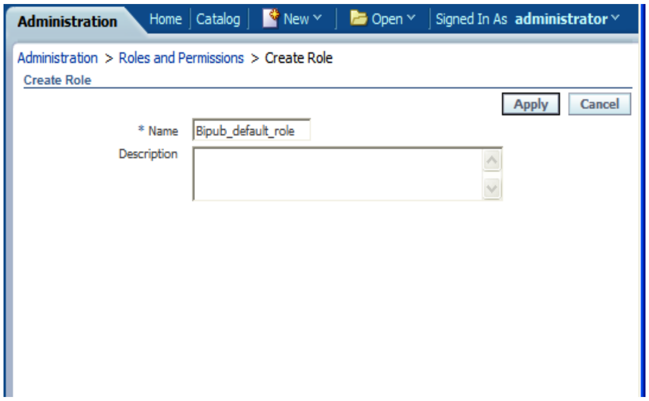
- Admin
- DemoFiles
- Reports
- Tools
- Users

29. Post install step: Create role Bipub_default_role.

   a. From the xmlpserver Administration screen, scroll down to Security Center and click Roles and Permissions.



   b. On the Roles and Permissions screen, click the Create Role button.


Administration
Home
Catalog
New
Open
Signed In As administrator
Administration > Roles and Permissions > Create Role
Create Role
Apply
Cancel

* Name Bipub_default_role

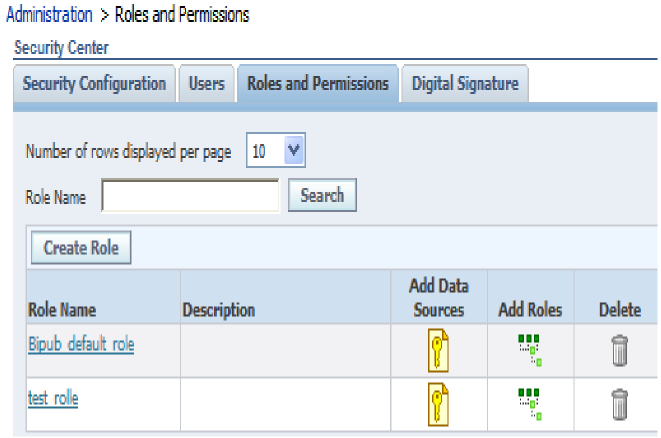
Description

   c. Create the Bipub_default_role. Enter in Create Role Section name of the role.

   d. When the information has been entered press Apply changes.

30. Post install step: Assign BiPub system roles to the newly created Bipub_default_role.

a. To assign BiPub system roles to the newly create Bipub_default_role, go to Security Center section and navigate to the Roles and Permissions screen:



Administration > Roles and Permissions

Security Center

Security Configuration | Users | Roles and Permissions | Digital Signature

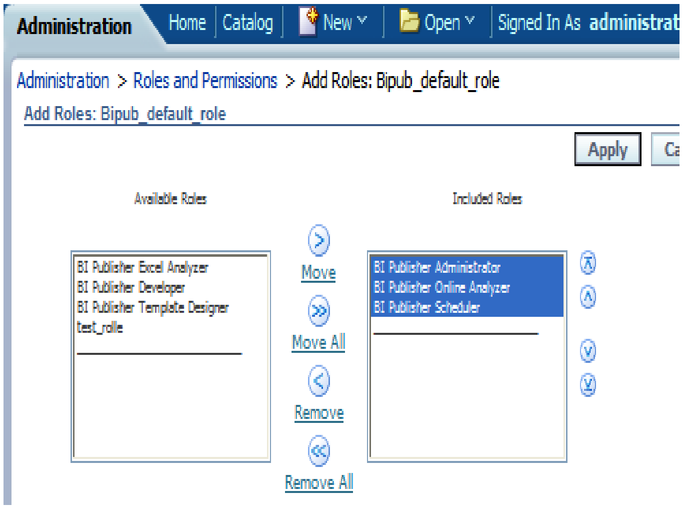
Number of rows displayed per page 10

Role Name Search

Create Role

Role Name	Description	Add Data Sources	Add Roles	Delete
Bipub_default_role				
test_role				

b. On the Roles and Permissions screen you should see the new role created: “Bipub_default_role” . Add multiple roles to the Bipub_Default_Role by pressing the corresponding green icon on the Add Roles column



Administration | Home | Catalog | New | Open | Signed In As administrat

Administration > Roles and Permissions > Add Roles: Bipub_default_role

Add Roles: Bipub_default_role

Apply | Ca

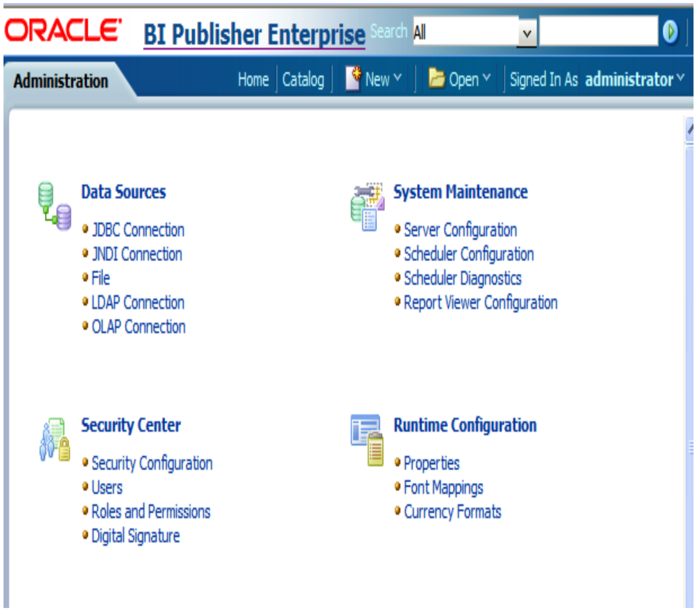
Available Roles:
- BI Publisher Excel Analyzer
- BI Publisher Developer
- BI Publisher Template Designer
- test_role

Included Roles:
- BI Publisher Administrator
- BI Publisher Online Analyzer
- BI Publisher Scheduler

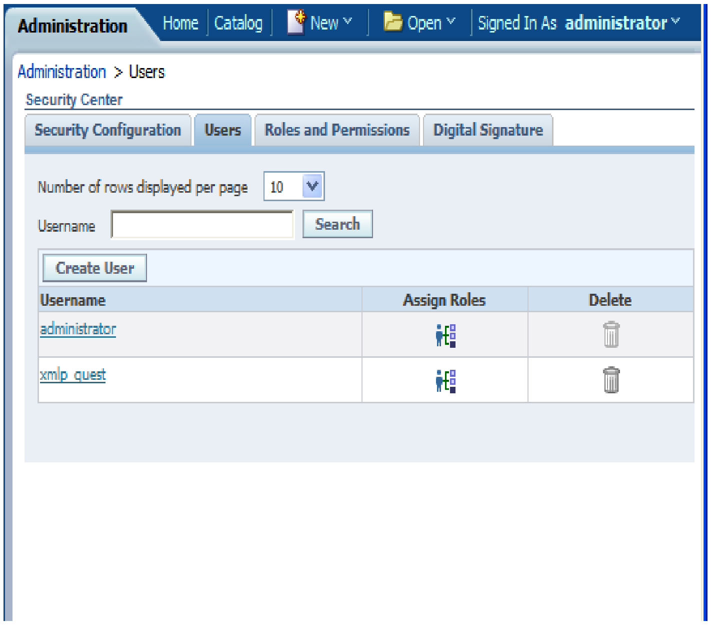
c. From the “Available Roles” panel, select the ones needed for your reports and move them to the “Included Roles” panel

d. Press the Apply button to save your changes.

31. Post install step: create Guest (XMLP_GUEST) user.



a. From the xmlpserver Administration screen scroll down to Security Center section and press Users to navigate to the next screen



Number of rows displayed per page 10

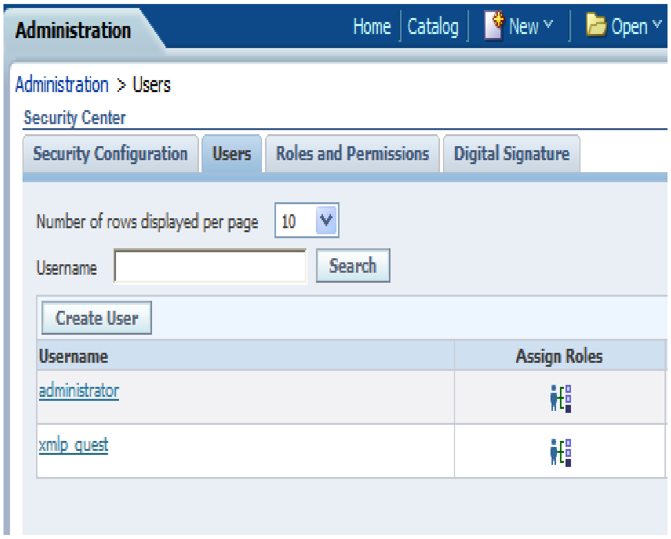
Username

Username	Assign Roles	Delete
administrator		
xmlp_guest		

b. Select the "Create User" button to create the "xmlp_guest" user and save the changes

32. Post install step: Adding the Bipub_default_role to XMLP_GUEST user.

a. Open the Users section:

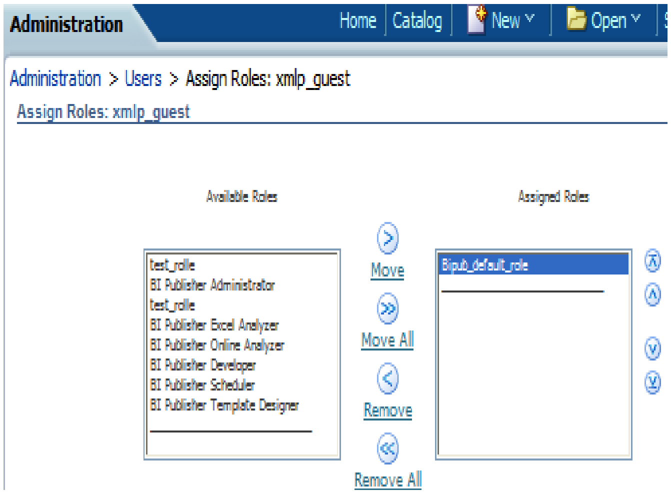


Number of rows displayed per page: 10

Username:

Username	Assign Roles
administrator	
xmlp_guest	

b. For xmlp_guest user, press on the "Assign Roles" icon to navigate to the next screen:



Available Roles

test_rolle
BI Publisher Administrator
test_rolle
BI Publisher Excel Analyzer
BI Publisher Online Analyzer
BI Publisher Developer
BI Publisher Scheduler
BI Publisher Template Designer

Assigned Roles

Bipub_default_role

c. On the Assign Roles screen, select the BiPub_default_role from the Available Roles panel to the "Assigned Roles" panel and press the Apply button to save your changes.

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, INSTALL_DIR/reports, and must be copied into the newly created BI server repository directory.

1. Change directory to the <INSTALL_DIR> /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_obiee/user_projects/domains/bifoundation_dom
   ain/config/bipublisher/repository/Guest/ "
   ```

Placement of Report Templates

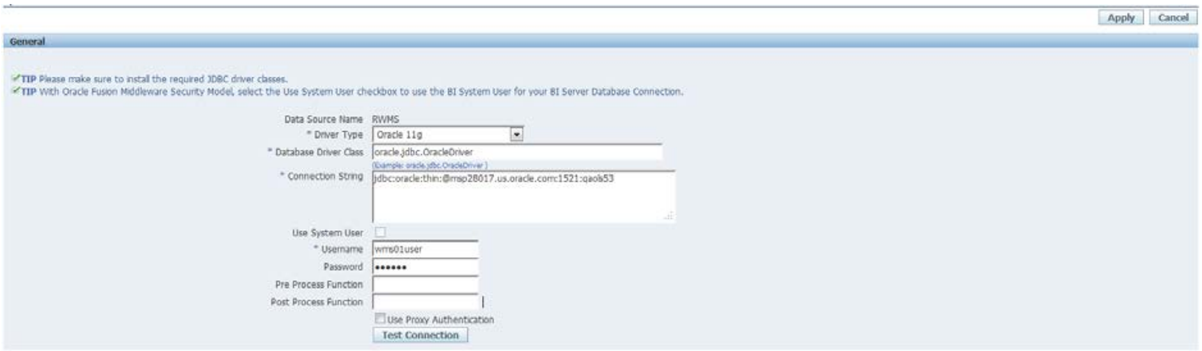
Directory	RWMS Repository	RWMS System Repository
asn_recv_package_audit_sys	✓	✓
asn_receiving_receipt_sys	✓	✓
best_before_date	✓	
bill_of_lading_sys	✓	✓
container_manifest_sys	✓	✓
gift_card_report_sys		✓
inventory_by_item	✓	
inventory_by_location	✓	
pack_slip_labels_sys		✓
pick_package_audit_sys	✓	✓
pts_containers_to_close_sys		✓
quality_audit_sys	✓	✓
receiving_receipt_sys	✓	✓
receiving_register	✓	
recv_package_audit_list_sys	✓	✓
return_to_vendor	✓	
rtv_advice_sys	✓	✓
unit_pick_group_sys		✓
unresolved_appointment	✓	

Directory	RWMS Repository	RWMS System Repository
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_task_history_sys		✓

Configuring the RWMS JDBC connection

Follow the below steps to configure JDBC connection for RWMS Data Source name. This datasource RWMS will be used for RWMS reports.

1. Log on with the default user id and passwords for BI Publisher using the administrative user and password configured previously.
2. Click the **Admin** tab. Select the JDBC Connection hyperlink in the Data Sources lists. The following screen will be displayed:



✓TIP Please make sure to install the required JDBC driver classes.
✓TIP With Oracle Fusion Middleware Security Model, select the Use System User checkbox to use the BI System User for your BI Server Database Connection.

Data Source Name RWMS
* Driver Type Oracle 11g
* Database Driver Class oracle.jdbc.OracleDriver
(Example: oracle.jdbc.OracleDriver)
* Connection String jdbc:oracle:thin:@msp28017.us.oracle.com:1521:qaols53
Use System User □
* Username wms01user
Password ••••••
Pre Process Function
Post Process Function
□ Use Proxy Authentication

3. Enter the appropriate details for the RWMS data source. Click Test Connection to test the connection on the screen once the data is entered.
   - Driver type is ORACLE 11g
   - Database driver class should be oracle.jdbc.OracleDriver.
   - Connection string is similar to this example: jdbc:oracle:thin:@msp28016:1521:qaols51
   - Syntax is jdbc:oracle:thin:@<hostname>:<port>:<dbsid>

Installing WebUtil Support Libraries

The RWMS application provides a means to upload or view files through a web browser and using its database as a storage mechanism. This means that the user can browse files in his local drives and select which ones to be uploaded and at the same time, download files from the database and save it in his local drive for viewing.

WebUtil is a small Java class library that is used for this purpose and will need the following configuration tasks to integrate it into the RWMS application environment.

To install WebUtil follow the instructions published by Oracle already located in a subdirectory of the base FORMS_PATH directory specified in the <INSTALLATION_NAME>.env file. The FORMS_PATH structure will be different for installation, but in every case it will specify a path similar to the following: /u00/webadmin/product/10.3.X/WLS_Forms/as_1/forms

Following the sample above the instruction documents will be located in the following directory:

```
/u00/webadmin/product/10.3.X/WLS_Forms/as_1/forms/doc/US/fmdevhlp/web_util.jar
```

The following steps below are a quick reference on what needs to need to be done for the installation and configuration of WebUtil for RWMS. It is a very high level summary of what needs to be done form the documentation packaged with the Oracle Forms installation. Some of the steps for doing this are also embedded in the sections below as well to assist in making sure nothing was missed.

When you complete the steps in the following section," Configuring Forms to use BI and WebUtil," be sure to verify that you have done the following:

- Executed the create_webutil_db.sql to create webutil database objects.
- Copied jacob.jar to forms/java folder
- Modified the <INSTALLATION_NAME>.env file to correct jacob.jar location in CLASSPATH.
- Compiled webutil.pll.
- Edited the webutil.cfg file and set transfer.database.enabled=TRUE .

Note: As a post install step to the webutil install - an additional update will need to be made to the jacob.jar due to enhanced java security requirements. The jacob.jar needs to be updated as follows: and updated it as follows:

- Unjar the jacob.jar
- Remove all the signature information from the manifest
- Add in the following lines of code to the manifest file:
- Permissions: all-permissions
- Codebase: *
- Jar the contents back up again
- Resign the jar.

Note: The jacob.jar file is not an Oracle supplied jar file, hence it requires signing with a trusted certificate from a Certificate Authority or a self signed certificate to function properly with webutil. Failure to sign the jacob.jar file with an RSA certificate will result in the application not working. The following link gives instructions on how to properly sign the jacob.jar. http://docs.oracle.com/javase/7/docs/technotes/guides/plugin/developer_guide/rsa_signing.html

Configuring Forms to use BI and WebUtil

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

Check the Configuration of formsweb.cfg

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

The file is typically found within the WebLogic domain configuration folder in the subdirectory ..../fmwconfig/servers/<SERVER_NAME>/applications/formsapp_11.1.2 /config directory and is named from the <INSTALLATION_NAME> value.

For example:

```
/u00/webadmin/product/10.3.X/WLS_Forms/user_projects/domains/ClassicDomain/config/
fmwconfig/servers/WLS_FORMS/applications/ formsapp_11.1.2/config/
```

The following is a sample from **formsweb.cfg.**

```
[webutil]
WebUtilArchive=frmwebutil.jar,jacob.jar
WebUtilLogging=off
WebUtilLoggingDetail=normal
WebUtilErrorMode=Alert
WebUtilDispatchMonitorInterval=5
WebUtilTrustInternal=true
WebUtilMaxTransferSize=16384
baseHTML=webutilbase.htm
baseHTMLjpi=webutiljpi.htm
archive=frmall.jar
lookAndFeel=oracle


[rwms14inst]
envfile=./rwms14inst/rwms14inst.env
width=950
height=685
separateFrame=true
lookAndFeel=Oracle
colorScheme=swan
archive=frmall.jar,rwms-icons.jar,frmwebutil.jar,jacob.jar
form=logon_scr.fmx
userid=/@rwms14inst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwms14inst_hh]
```

```
envfile=./rwms4inst/rwms14inst.env
width=100%
height=100%
separateFrame=false
form=hh_intro_s.fmx
lookAndFeel=Oracle
Logo=false
colorScheme=swan
background=false
ShowMenuBar=false
ShowStatusBar=false
baseHTMLjinitiator=basejini.htm
baseHTML=base.htm
otherparams=term=/u00/webadmin/product/10.3.X/WLS_Forms/user_projects/domains/Clas
sicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2
/config/rwms14inst/fmrweb.res usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0 marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=/@rwms14inst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwms14inst_tm]
envfile=./rwms4inst/rwms14inst.env
width=100%
height=100%
separateFrame=false
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/product/10.3.X/WLS_Forms/user_projects/domains/Clas
sicDomain/config/fmwconfig/servers/WLS_FORMS/applications/ formsapp_11.1.2
/config/rwms14inst/fmrweb.res usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0 marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=/@rwms14inst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwms14inst_wr]
envfile=./rwms4inst/rwms4inst.env
width=100%
height=100%
separateFrame=false
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/product/10.3.X/WLS_Forms/user_projects/domains/Clas
sicDomain/config/fmwconfig/servers/WLS_FORMS/applications/ formsapp_11.1.2 /c
```

```
onfig/rwms14inst/fmrweb.res usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0 marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=/@rwms14inst
#ssoMode=true
#ssoDynamicresourceCreate=true
```

Configuring the Forms Installation Environment Files

This section outlines the entries that need to be updated in the <INSTALLATION_NAME>.env file. The file is typically found within the WebLogic domain configuration folder in the subdirectory ..../fmwconfig/servers/<SERVER_NAME>/applications/formsapp_11.1.2 /config directory and is named from the <INSTALLATION_NAME> value. For example:

```
/u00/webadmin/product/10.3.X/WLS_Forms/user_projects/domains/ClassicDomain/config/
fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2 /config/rwms14inst/
```

Field Title	RWMS_REPORTS_URL
Field Description	This field points to BI Publisher URL location for the RWMS reports repository.
Example	http://<BI_PublisherURL>:<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 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.
Example	/u00/webadmin/product/10.3.6/WLS_Forms/user_projects/domains /ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/ formsapp_11.1.2 /spn4wj /war/rwms/reptemp/

Field Title	RWMS_REPORT_END_POINT_URL. The installer will update a blank property. Please update the property with the URL
Field Description	This field points the BI Publisher RWMS Reports end point URL
Example	http://<BI_PublisherURL>:<Port>/xmlpserver/services /v2/ReportService?wsdl

Field Title	RWMS_SCHEDULE_END_POINT_URL
Field Description	This field points the BI Publisher RWMS scheduler end point URL.The installer will update a blank property. Please update the property with the URL
Example	http://<BI_PublisherURL>:<Port>/xmlpserver/services /v2/ReportService?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 mechanism is in use. NOTE: The RWMS installer by default does not even 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: 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	This is set to the Java wallet alias that was created 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. 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.
Example	Wallet Example: <schema_owner>_<db_sid> Non wallet usage: wms01/retail@mydb
Field Title	RWMS_WALLET_PATH
Field Description	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.
Example	/projects/rwms14/extras/javawallet

Field Title	RWMS_WALLET_PARTITION
Field Description	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.
Example	rwms14inst

Field Title	RWMS_WALLET_LOGON
Field Description	This variable is used by the application to determine if it should consider theRWMS_BI_USER – RWMS_BI_PWD – and RWMS_DB_CONNECT variables as wallet aliases or string values. NOTE: 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.
Example	TRUE or FALSE

Field Title	RWMS_SYS_REPORTS_DIR
Field Description	This variable is set to the directory within BI Publisher in which the reports reside for process dependent reports. Definition of reports that go into this file will be illustrated later.
Example	/Guest/RWMS/RWMS SYSTEM REPOSITORY/

Field Title	RWMS_FORMS_SERVER
Field Description	The port needs to reflect the port the application URL will be using. The rest of it is the first part of the forms application URL.
Example	http://mspdv351:8888

Field Title	RWMS_REPTEMP_ALIAS
Field Description	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	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	/u00/projects/rwms14.0/devlinf/extras

Field Title	RF_LAUNCH_VALUE_ALIAS
Field Description	This variable is used by the forms application server to find the RF launch screen modules and render them to the user.
Example	/forms/rwms/rf_launch/

Create a directory called extras in the top-level directory of your RWMS installation (referenced in this guide as MMHOME. For reference, the forms/bin directory will also be located in there.)

Example:

```
/u00/oretail/rwms14.0/tst/extras
```

Pick up the following files and place them in the extras directory that was just created. The files will be located in the rwms installers <STAGING_DIR>/rwms/installer/rwms14/ Enhanced_Navigation_and_Dashboards/Client/lib/ directory.

- httputil.jar
- wmssecurity.jar
- bihelper.jar

Field Title	CLASSPATH
Field Description	You will need to add the jars listed below: **httputil.jar** Should exist in the extras directory that was manually created. **bihelper.jar** Should exist in the extras directory that was manually created. **wmsSecurity.jar** Should exist in the extras directory that was manually created. The below jars are also to be added in the classpath. **jrf.jar** <ORACLE_HOME>/oracle_common/modules/oracle.jrf_11.1.1/jrf.jar **javax.persistence_1.0.0.0_2-0-0.jar -** <ORACLE_HOME>/oracle_common/modules/javax.persistence_1.0.0.0_2-0-0.jar **xmlparserv2.jar -** <ORACLE_HOME>/oracle_common/modules/oracle.xdk_11.1.0/xmlparserv2.jar **xml.jar -** <ORACLE_HOME>/oracle_common/modules/oracle.xdk_11.1.0/xml.jar **orai18n-mapping.jar -** <ORACLE_HOME>/oracle_common/modules/oracle.nlsrtl_11.1.0/orai18n-mapping.jar **com.oracle.toplink_1.1.0.0_11-1-1-6-0.jar -** <ORACLE_HOME>/modules/com.oracle.toplink_1.1.0.0_11-1-1-6-0.jar **org.eclipse.persistence_1.2.0.0_2-3.jar -** <ORACLE_HOME>/modules/org.eclipse.persistence_1.2.0.0_2-3.jar **com.bea.core.antlr.runtime_2.7.7.jar -** <ORACLE_HOME>/modules/com.bea.core.antlr.runtime_2.7.7.jar **weblogic.jar** <ORACLE_HOME>/wlserver_10.3/server/lib/weblogic.jar

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

```
…
/u00/webadmin/product/10.3.X/WLS/oracle_common/modules/oracle.jrf_11.1.1/jrf.jar:/
u00/webadmin/product/10.3.X/WLS/wlserver_10.3/server/lib/weblogic.jar:/u00/webadmi
n/product/10.3.X/WLS/modules/com.oracle.toplink_1.0.0.0_11-1-1-5-
0.jar:/u00/webadmin/product/10.3.X/WLS/modules/org.eclipse.persistence_1.1.0.0_2-
1.jar:/u00/webadmin/product/10.3.X/WLS/modules/javax.persistence_1.0.0.0_2-0-
0.jar:/u00/webadmin/product/10.3.X/WLS/oracle_common/modules/oracle.xdk_11.1.0/xml
parserv2.jar:/u00/webadmin/product/10.3.X/WLS/oracle_common/modules/oracle.xdk_11.
1.0/xml.jar:/u00/webadmin/product/10.3.X/WLS/oracle_common/modules/oracle.nlsrtl_1
1.1.0/orai18n-
mapping.jar:/u00/webadmin/product/10.3.X/WLS/Oracle_FRHome1/forms/j2ee/frmsrv.jar:
/u00/webadmin/product/10.3.X/WLS/Oracle_FRHome1/jlib/ldapjclnt11.jar:/u00/webadmin
/product/10.3.X/WLS/Oracle_FRHome1/jlib/debugger.jar:/u00/webadmin/product/10.3.X/
WLS/Oracle_FRHome1/jlib/ewt3.jar:/u00/webadmin/product/10.3.X/WLS/Oracle_FRHome1/j
lib/share.jar:/u00/webadmin/product/10.3.X/WLS/Oracle_FRHome1/jlib/utj.jar:/u00/we
badmin/product/10.3.X/WLS/Oracle_FRHome1/jlib/zrorg.eclipse.persistence_1.1.0.0_2-
1.jaras/wmsSecurity.jar:/u00/projects/rwms14.0/devlinf/extras/bihelper.jar:/u00/we
badmin/product/10.3.X/WLS/modules/com.bea.core.antlr.runtime_2.7.7.jar:/u00/projec
ts/rwms14.0/devlinf/extras/httputil.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://msp52266.us.oracle.com:29704/xmlpserver/servlet/r
eport?f=/Guest/RWMS/RWMS+REPOSITORY
RWMS_REPORTS_TEMP=/u00/webadmin/product/10.3.X/WLS/user_projects/domains/
ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_11.1.2/spn4wj/war/r
wms/reptemp

ORACLE_RWMS_EXTRAS=/u00/projects/rwms14.0/devlinf/extras
RWMS_FORMS_SERVER=http://redevlv0081.us.oracle.com:9001
RWMS_REPTEMP_ALIAS=/forms/rwms/reptemp/
RF_LAUNCH_VALUE_ALIAS=/forms/rwms/rf_launch/

RWMS_WALLET_PATH= /projects/rwms14/extras/javawallet
RWMS_WALLET_PARTITION=rwms14inst
RWMS_WALLET_LOGON=TRUE
RWMS_BI_PWD=BI_ALIAS
RWMS_DB_CONNECT=wms01_mydb
RWMS_REPORT_END_POINT_URL=http://msp52266.us.oracle.com:29704/xmlpserver/
services/v2/ReportService?wsdl
RWMS_SCHEDULE_END_POINT_URL=http://msp52266.us.oracle.com:29704/xmlpserve
r/services/v2/ScheduleService?wsdl
```

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=/u00/projects/rwms14.0/devlinf/forms/wallet
#RWMS_WALLET_PARTITION=rwms14inst
```

```
RWMS_WALLET_LOGON=FALSE
RWMS_BI_USER=retail.user
RWMS_BI_PWD=welcome1
RWMS_DB_CONNECT=wms01/retail@mydb
```

Adding Radio Frequency Launch Configurations

The RF Launch Screen is used as an entry point for RF devices to the RWMS application when users first log on to the terminal server. It is how the system knows what type of device is accessing the RWMS application as well as set system variables according to the type of equipment being used. The RWMS installer already sets this up with corresponding formsweb.cfg and .env file entries. It is necessary to know how to set these up however so that other devices can be added.

File Placement

To place files, complete the following steps.

> **Note:** Instructions for placing files are included in the steps for installing BI Publisher. If you have already completed BI Publisher installation, you can skip this step.

1. On the application server in the forms application server base directory you will need to create the following directory

   /rwms/rf_launch/

   Example:
   /u00/webadmin/product/10.3.6/WLS_Forms/user_projects/domains/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_11.1.2 / spn4wj/war/rwms/rf_launch
2. Place the following files in the directory that was created during this step.
   - close.htm

   > **Warning:** Do not open the close.htm file. It is designed to automatically shut down the computer that opens it. Use a text editor to view the contents as needed.

   - oracle_logo.jpg
   - rwms_rf_menu.htm

Configure .env file Variables

See notes above for .env file updates.

Update the RF Launch Modules

The close.htm, oracle_logo.jpg, and rwms_rf_menu.htm all need to be placed in the forms application server base directory/rwms/rf_launch. The rwms_rf_menu.htm will then need to be configured to point to launch the correct URL's based on the names of the URL's in the formsweb.cfg file. The variables that need to be changed are listed below.

- var hh_device = "http://<Server_Name>:<port>/forms/frmservlet?config=<INSTALLATION_NAME>_hh";
- 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";

8

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 (RWMS14DEV). 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.

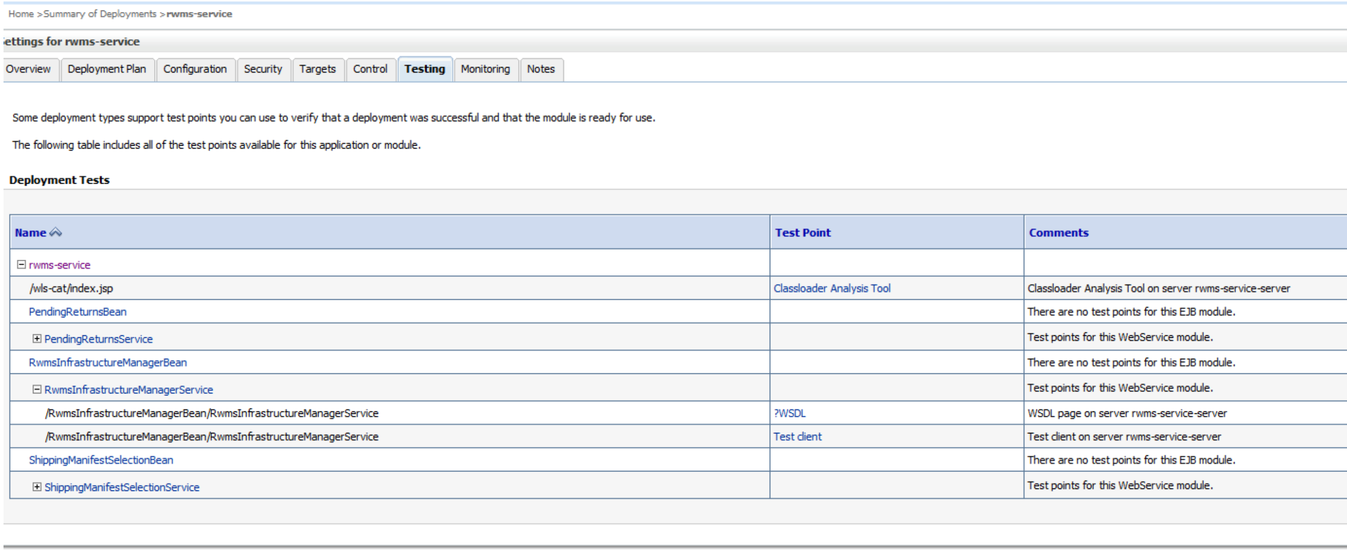
Deploy RWMS Service EAR File

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 Deployment Archive browse button. Click the Upload button.
6. Go to the location of the rwms-service.ear file. This would be <STAGING_DIR>/rwms/installer/rwms14/webservice_objects/provider/ear
7. Select the rwms-service.ear file.
8. Click **Next**. Make sure that the radio button for rwms-services.ear is selected.
9. Click **Next** again. Make sure that Install this deployment as an application is selected.
10. Click **Next** again and select the server created in Step 2.
11. Click **Next**. Click **Finish** to return to the deployments page. You should see rwms-service in the list of deployments.

12. 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 will enable the Start button. Click **Start**. Select servicing all requests.

13. To test the deployment, click on the application. Click the testing tab.



Some deployment types support test points you can use to verify that a deployment was successful and that the module is ready for use.

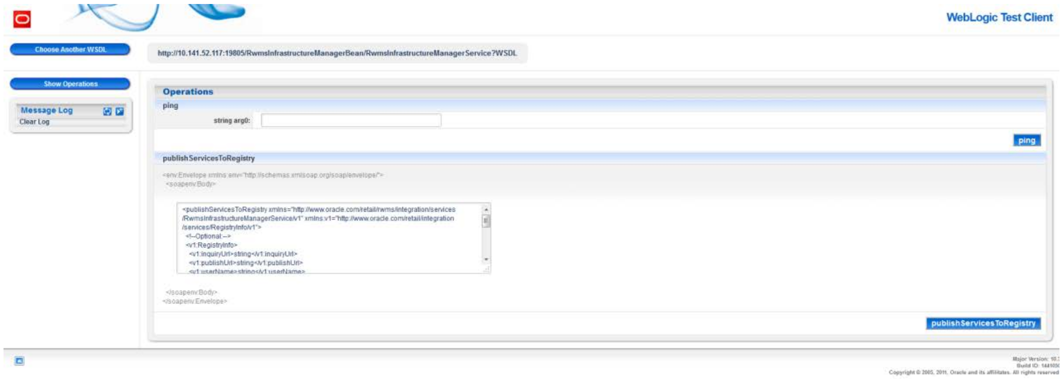
The following table includes all of the test points available for this application or module.

Deployment Tests

Name	Test Point	Comments
⊟ rwms-service		
/wls-cat/index.jsp	Classloader Analysis Tool	Classloader Analysis Tool on server rwms-service-server
PendingReturnsBean		There are no test points for this EJB module.
⊞ PendingReturnsService		Test points for this WebService module.
RwmsInfrastructureManagerBean		There are no test points for this EJB module.
⊟ RwmsInfrastructureManagerService		Test points for this WebService module.
/RwmsInfrastructureManagerBean/RwmsInfrastructureManagerService	?WSDL	WSDL page on server rwms-service-server
/RwmsInfrastructureManagerBean/RwmsInfrastructureManagerService	Test client	Test client on server rwms-service-server
ShippingManifestSelectionBean		There are no test points for this EJB module.
⊞ ShippingManifestSelectionService		Test points for this WebService module.

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 temporary 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.

14. Expand one of the four web services. Click the ?WSDL and Test Client links to test. For the test client you should see a screen similar to the following:



http://10.141.52.117:19805/RwmsInfrastructureManagerBean/RwmsInfrastructureManagerService?WSDL

string arg0:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
   <publishServicesToRegistry xmlns="http://www.oracle.com/retail/rwms/integration/services
   /RwmsInfrastructureManagerService/v1" xmlns:v1="http://www.oracle.com/retail/integration
   /services/RegistryInfo/v1">
    <!--Optional:-->
    <v1:RegistryInfo>
     <v1:inquiryUrl>string</v1:inquiryUrl>
     <v1:publishUrl>string</v1:publishUrl>
     <v1:userName>string</v1:userName>
 </soapenv:Body>
</soapenv:Envelope>
```

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:

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

Refer to Oracle Retail Warehouse management system Security Guide along with RTG Security Guide for more details.

9

RWMS Patching Procedures

Patching Changes in RWMS 14.0

The patching process for Oracle Retail Warehouse Management System (RWMS) 14.0 has been substantially revised from past releases. New automation has been created 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.

In addition to new metadata on the environment and the utilities to automate patching activities, some changes have been made to the approach when applying patches.

Many enhancements have been made in RWMS 14.0 to make installing and updating RWMS installations easier and reduce opportunities for mistakes. All of the utilities and infrastructure described in this and subsequent sections are new with RWMS 14.0.

The following patching changes have been made for RWMS 14.0:

- New utilities have been created to automate the application of some types of Oracle Retail patches to environments.
- Unified patches are being utilized so that a single patch can be applied against Database and Forms.
- Database and Environment manifests allow you to track versions of files at a module level.
- The centralized configuration distinguishes installation types (Database or Forms).
- The patch inventory tracks the patches applied to an environment.

Some of these changes add additional considerations to patching and maintaining an RWMS environment. Additional details on these considerations are found in later sections.

Patch Concepts

During the lifecycle of an RWMS 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 14.0.0 or 14.0.1 release, or incremental, such as a hot fix for just a few modules. Patches may contain updates for some or all components of an RWMS installation including database, and forms. 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 forms changes, the patch would need to be applied to both the database server and the forms server.

Some components of RWMS do not utilize the Oracle Retail Patch Assistant framework, including RWMS Reports and the Enhanced Dashboards and Navigation application. Patches against these components should be applied manually, following the instructions included with each patch.

The top-level directory for the installation of RWMS is referred to as the MMHOME. Underneath MMHOME are all of the files related to that application 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 MMHOME. For example, the compiled RWMS forms.

Patch Structure

An Oracle Retail 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 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 the Oracle Retail Patch Assistant.

Manifest Files

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

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)

Oracle Retail Patch Assistant is the utility used to apply patches to an Oracle Retail Warehouse Management System installation. It is used in the background by the RWMS 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 forms will recompile the form files with each patch. Merging the patches into a single patch before applying them will allow forms to be recompiled only once.

Oracle Retail Compile Patch (ORCompile)

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

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 14.0.1, 14.0.2, 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.

Version Tracking

The new patching infrastructure for RWMS 14.0 tracks version information for all files involved with an RWMS installation. The MMHOME now contains files which track the revision of all files within the MMHOME including batch, forms, database and other files. In addition, records of database scripts that have been applied to the RWMS objects are kept within each database schema.

Environment Configuration

A new configuration file in $MMHOME/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, 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 Database Scripts

The MMHOME location of an RWMS installation contains all of the files associated with that installation. This includes Forms files as with previous versions and also includes all database scripts. This allows database objects to be easily reloaded including their 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
  - SP jars and packages
  - Form Elements
- All forms files will be recompiled

When applying incremental patches, only changed files will be reloaded.

Backups

Before applying a patch to an environment, it is extremely important to take a full backup of both the MMHOME filesystem and the RWMS database. Although Oracle Retail Patch Assistant 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 $MMHOME/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 Forms and Batch files are saved before recompilation. These backups may be created during patches:

- Forms 'forms' directory in $MMHOME/forms/bin-<timestamp>

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

Maintenance Considerations

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

Apply all Patches with Installer or Oracle Retail Patch Assistant

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

Password Changes

Oracle wallets are used to protect the password credentials for connecting to the RWMS database schemas. This includes all database schemas used during an install: the RWMS owning schema, the RWMS_ADF_USER schema, and the RWMS runtime user schema. 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 $MMHOME/orpatch/rwms_wallet for database installations, and $MMHOME/orpatch/rwms_wallet_app for forms installations. The wallet alias for each schema will be <username>_<dbname>. Standard mkstore commands can be used to update the password.

For example:

```
mkstore -wrl $MMHOME/orpatch/rwms_wallet –modifyCredential rwms_rwmsdb rwms01
rwmspassword
```

This command will update the password for the RWMS01 user to 'rwmspassword' in the alias 'rwms_rwmsdb'.

Infrastructure Directory Changes

The MMHOME/orpatch/config/env_info.cfg file contains the path to the database ORACLE_HOME on RWMS database installations and to the WebLogic Forms and Reports ORACLE_HOME and ORACLE_INSTANCE on Forms installations. If these paths change, the related configuration variables in the env_info.cfg file must be updated.

DBManifest Table

The table dbmanifest within the RWMS schemas is used to track the database scripts which have been applied to the database. 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.

MMHOME relationship to Database

The importance of the MMHOME to an RWMS installation has increased due to the additional metadata and historical information contained within it. If a database is moved or copied, the MMHOME related to it should be copied or moved at the same time.

Patching Operations

Running Oracle Retail Patch Assistant

Oracle Retail Patch Assistant (ORPatch) is used to apply patches to an Oracle Retail Warehouse Management System 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 RWMS 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 RWMS forms, 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 an RWMS 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 MMHOME filesystem and the RWMS database. Although Oracle Retail Patch Assistant 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 $MMHOME/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/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/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

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 $MMHOME/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 $MMHOME/stage/123456
analyze	Tells ORPatch to analyze a patch, requires the –s option Example: orpatch analyze -s $MMHOME/stage/123456
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 $MMHOME/support directory to contain it. Requires the –expname option.

Action	Description
diffmetadata	Tells ORPatch to compare all metadata from the current environment with metadata exported from some other environment. Requires the –expname option.
revert	Tells ORPatch to revert the files related to a patch, requires the –s option Example: orpatch revert –s $MMHOME/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.
-selfonly	apply analyze	Only apply or analyze changes in a patch that relate to orpatch itself. This is useful to apply 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 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.
- Saves details of the files that will be impacted in $MMHOME/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 RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.
   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```
3. Create a staging directory to contain the patch, if it does not already exist.
   ```
   mkdir -p $MMHOME/stage
   ```
4. Download the patch to the staging directory and unzip it.
5. Execute orpatch to analyze the patch.
   ```
   orpatch analyze -s $MMHOME/stage/123456
   ```
6. Repeat the patch analysis on all servers with RWMS installations for this environment.
7. Evaluate the list of impacted files.

For more information on registering 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 RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.
   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```
3. Set the DISPLAY environment variable if the patch contains Forms.
   ```
   export DISPLAY=localhost:10.0
   ```

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

4. Create a staging directory to contain the patch, if it does not already exist.
   ```
   mkdir -p $MMHOME/stage
   ```
5. Download the patch to the staging directory and unzip it.
6. Review the README.txt included with the patch. If manual steps are specified in the patch, execute those steps at the appropriate time.
7. Shutdown applications.

8. Execute ORPatch to apply the patch.

   ```
   orpatch apply -s $MMHOME/stage/123456
   ```

9. After ORPatch completes, review the log files in $MMHOME/orpatch/logs.
10. Repeat the patch application on all servers with RWMS installations for this environment.
11. Restart applications.

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 $MMHOME/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 $MMHOME/orpatch/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 RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.

   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```

3. Execute orpatch to list the inventory.

   ```
   orpatch 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.

Exports of environment metadata are always done to the $MMHOME/support directory. When exporting metadata, you must specify the –expname argument and define the name that should be given to the export. The name is used for the directory within $MMHOME/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 RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.

   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```

3. Execute orpatch to export the metadata.

   ```
   orpatch exportmetadata –expname test_env
   ```

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

> **Note:** The $MMHOME/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 metadata of two environments. ORPatch provides a capability to compare metadata of the current environment with the exported metadata of another environment. 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 $MMHOME/orpatch/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 $MMHOME/support directory. The metadata should exist in the same structure that it was originally exported in. For example, if the metadata was exported to $MMHOME/support/test_env on another system, it should be placed in $MMHOME/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 $MMHOME/support/dev directory, you could run "orpatch 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 $MMHOME/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 orpatch exportmetadata.
2. Transfer the metadata zip from the other system to $MMHOME/support.
3. Log in as the UNIX user that owns the RWMS installation.
4. Set the MMHOME environment variable to the top-level directory of your RWMS installation.

   ```
   export MMHOME=/u00/oretail/rwms14.0/dev
   ```
5. Unzip the metadata zip file.

   ```
   unzip test_env.zip
   ```
6. Execute orpatch to compare the metadata.

   ```
   orpatch diffmetadata –expname test_env –srcname dev_env
   ```

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

> **Note:** The $MMHOME/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.

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.

To revert a patch, perform the following steps:

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

   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```
3. Identify the backup directory in $MMHOME/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

4. Execute orpatch to revert the environment using the contents of the backup directory
```
orpatch revert –s $MMHOME/backups/backup-11232013-152059
```
5. Restore the database from backup if the patch made database changes
6. Use the orcompile script to recompile forms if the patch included RWMS forms files
```
orcompile –a RWMS –t FORMS
```

Merging Patches

When patches are applied individually some Oracle Retail Patch Assistant tasks such as compiling forms files or compiling database invalid objects 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.

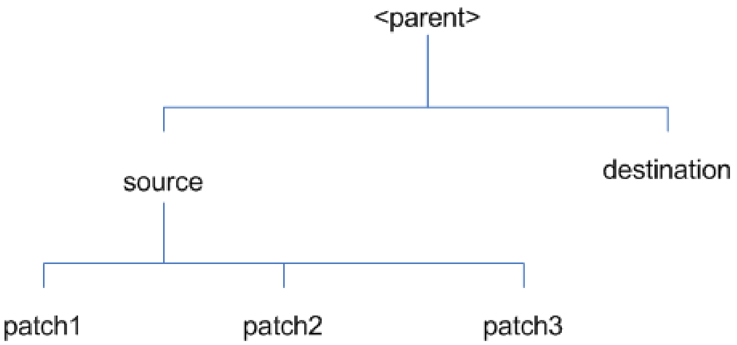
Source and Destination Directories

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

Argument	Required	Description
-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.

Running the ORMerge Utility

To merge patches, perform the following steps:

1. Log in as the UNIX user that owns the RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.
   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```
3. Create a staging directory to contain the patches.
   ```
   mkdir –p $MMHOME/stage/merge/src
   ```
4. Download the patches to the staging directory and unzip them so that each patch is in a separate subdirectory.
5. 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.
6. Create a destination directory to contain the merged patches.
   ```
   mkdir -p $MMHOME/stage/merge/dest
   ```
7. Execute ormerge to merge the patches.
   ```
   ormerge -s $MMHOME/stage/merge/src –d $MMHOME/stage/merge/dest –name
   merged_patch
   ```

The merged patch can now be applied as a single patch to the RWMS environment using ORPatch.

Compiling Application Components

In some cases it may be desirable to recompile RWMS Forms 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 Oracle Retail Patch Assistant 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
RWMS	FORWMS	Compile RWMS Forms
RWMS	DB	Compile invalid database objects in the primary RWMS schema
RWMS	DB-ADF	Compile invalid database objects in the RWMS_ADF_USER schema

Running the ORCompile Utility

To compile files, perform the following steps:

1. Log in as the UNIX user that owns the RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.
   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```
3. Execute orcompile to compile the desired type of files.
   ```
   orcompile –a <app> -t <type>
   ```

ORCompile Examples

Compile RWMS Forms.

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

Compile invalid objects in the RWMS owning schema.

```
orcompile -a RWMS -t DB
```

Customization

Patching Considerations with Customized RWMS Files and Objects

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

General Guidelines

It is always preferred to customize RWMS by extension rather than by direct modification. For example, adding new database objects and forms rather than modifying existing Oracle Retail objects and forms.

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 RWMS schema owner and other users to access your objects, and use synonyms and grants to allow your customizations to access RWMS objects. A separate schema will ensure that your customizations are segregated from base Oracle Retail code.

Oracle Retail Patch Assistant 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 RWMS schema could result in failures to apply patches as changes to RWMS 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 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.

Analyze Patches when Customizations are Present

Whenever you have customized RWMS 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 Oracle Retail Patch Assistant 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 $MMHOME/orpatch/bin/orcustomreg script.

The orcustomerg 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.
-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 MMHOME. 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 RWMS installation.
2. Set the MMHOME environment variable to the top-level directory of your RWMS installation.
   ```
   export MMHOME=/u00/oretail/rwms14.0/tst
   ```
3. Execute orcustomreg script to register the desired file(s).
   ```
   orcustomreg –register –f <file>
   ```

Examples of using the orcustomreg Script

Register $MMHOME/dbsql_rwms/Packages/Source/zone.pkg as customized.

```
orcustomreg -f dbsql_rwms/Packages/Source/zone.pkg
```

Unregister customizations for $MMHOME/dbsql_rwms/Triggers/Source/TR_WAVE.trg

```
orcustomreg –unregister –f $MMHOME/dbsql_rwms/Triggers/Source/TR_WAVE.trg
```

Bulk register several files as customized.

```
echo "$MMHOME/dbsql_rwms/Procedures/Source/v_zone.osp" > custom.txt
echo "$MMHOME/dbsql_rwms/Procedures/Source/v_pfl_item_id.osp" >> custom.txt
echo "$MMHOME/dbsql_rwms/Procedures/Source/v_time.osp" >> custom.txt
orcompile –bulk custom.txt
```

List all files registered as customized.

```
orcustomreg –list
```

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 Warehouse Management System 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 RWMS database schema, while a separate action is responsible for compiling RWMS forms. These actions are enabled and disabled within the environment configuration file, allowing ORPatch to determine what types of changes to apply to each installation of RWMS.

ORPatch Actions for Oracle Retail Warehouse Management

Order	Action Name	Description
1	DBSQL_RWMS	Loads RWMS database objects into the primary RWMS schema
2	RWMSFORMS	Compiles RWMS Forms
3	DBSQL_RWMSADF	Loads database objects into the RWMS_ADF_USER schema
4	DBSQL_RWMSUSER	Determines schema name of RWMS runtime user, for privilege grants and synonyms on RWMS tables.

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. Currently no actions implement activities in this phase.

Restart Phase Number	Phase Name	Description
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 MMHOME/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.

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 MMHOME. 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_RWMS 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.

Example Custom Hook Definitions

A shell script that is executed prior to the Pre-Action phase of RWMS Forms:

```
RWMSFORMS_PREACTION_START=/u00/oretail/prepare_permissions.sh
```

A shell script that is executed after Forms files are copied into the MMHOME:

```
RWMSFORMS_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 MMHOME/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 MMHOME/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 manually update the dbmanifest table to record that a specific script does not need to be run. Before doing this ensure that all statements contained in the script have been completed.

To manually update the dbmanifest table perform these steps:

1. Find the failing script within the MMHOME.

   ```
   find $MMHOME -name <sql_file_name>
   ```

2. Starting with the directory the sql script is located in, look for a file called env_manifest.csv. If the file does not exist in the directory with the SQL script, back up one directory until you find it. Normally the env_manifest.csv will be in $MMHOME/dbsql_rwms or $MMHOME/dbsql_rwmsadf.
3. Get the line for the sql file from the env_manifest.csv.

   ```
   grep <sql_file_name> $MMHOME/<action>/env_manifest.csv
   ```

4. The output should be a single line with comma-separated values including: the relative destination file name, a source file name, the file revision, a file type, and a patch name.
5. Create an insert statement that looks like the following:

   ```
   insert into dbmanifest(file_name,version,file_type,patch_name,applied_date)
   values(‘<dest file name>’,’<revision number>’,’<file type>’,’<patch
   name>’,sysdate);
   ```

   For example:

   ```
   insert into dbmanifest(file_name,version,file_type,patch_name,applied_date)
   values('Install_DDL/rwms_ddl.sql',
   ‘700139’,'rwms_db_ddl','RWMS_14_0_0',sysdate);
   ```

6. Connect to the Retail database as the schema used by the failing action.
7. Run the insert statement against the dbmanifest table.
8. Select the data to ensure that it looks correct.
9. Commit your changes.

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 relative filename does not match the failing script exactly.
- The file type is not correct.
- 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 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 MMHOME/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 RWMS 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.

Patching Prior to First Install

In some situations, it may be necessary to apply a patch to RWMS 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 the RWMS installation files prior to starting an installation:

1. Unzip the RWMS installation files to a staging area.

   > **Note:** The following steps assume the files are in /media/rwms14.0

2. Create a logs directory within the RWMS installation staging area

   ```
   mkdir /media/rwms14.0/rwms/installer/rwms14/Build/orpatch/logs
   ```

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

   ```
   grep “PATCH_NAME=” /media/rwms14.0/rwms/installer/rwms14/patch_info.cfg
   ```

   Output Example:

   PATCH_NAME=RWMS_14_0_0_0

4. Create a directory that will contain the patch that must be applied, next to the directory with the RWMS 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 MMHOME to point within the RWMS installation.

   ```
   export MMHOME=/media/rwms14.0/rwms/installer/rwms14/Build
   ```

7. Ensure the ORMerge shell script is executable.

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

8. Run ORMerge to apply the patch to the RWMS installation, using a –name argument that is the same as what was found in step 3.

   ```
   $MMHOME/orpatch/bin/ormerge -s /media/patch -d
   /media/rwms14.0/rwms/installer/rwms14 –name RWMS_14_0_0_0 –inplace
   ```

Note: The –inplace argument is critical to ensure that the patching replaces files in the rwms14 directory.

9. Unset the MMHOME environment variable.

```
unset MMHOME
```

At this point, the RWMS 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/rwms14.0/rwms/installer/rwms14/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.

A

Appendix: Oracle Database 11g Parameter File

```
###############################################################################
# Oracle 11.2.0.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.
# ------------------------------------------------------------------------
# MAINTENANCE LOG
#
# Date     By          Parameter           Old/New         Notes
# +------+ +---------+ +-----------------+ +-------------+ +-------------+
#
#
###############################################################################
# -----------------------------------------------------------------------------
# The policy is to give 60% for sga and 40% for PGA out of Memory Target at
startup
# -----------------------------------------------------------------------------
memory_target                          = 3000M
# -----------------------------------------------------------------------------
audit_file_dest              = <admin_path>/adump
compatible            = 11.2.0
control_files                          = (<datafile_path>/control01.ctl
                                         ,<datafile_path>/control02.ctl)
db_block_size            = 8192      # Default is 2k; adjust before db creation,
cannot change after db is created
db_file_multiblock_read_count        = 16        # Platform specific (max io
size)/(block size)
db_name               = SID
diagnostic_dest                        = '<diag_path>'
java_pool_size           = 100M
job_queue_processes          = 5        # Oracle Retail required; number of
cpu's + 1
local_listener               =
"(ADDRESS=(PROTOCOL=TCP)(HOST=<hostname>)(PORT=1521))"
nls_calendar                           = GREGORIAN
nls_date_format              = DD-MON-RR # Oracle Retail required; if RDW
database see later entry for proper format
nls_language             = AMERICAN  # Default
nls_numeric_characters       = ".,"      # Should be explicitly set to ensure all
users/batch get the same results
nls_sort              = BINARY    # Should be explicitly set to ensure all
sessions get the same order
nls_territory            = AMERICA   # Default
open_cursors             = 900       # Oracle Retail required (minimum=900);
default is 50
plsql_optimize_level                   = 2        # 10g change; use this setting
to optimize plsql performance
```

```
processes             = 2000       # Max number of OS  processes that can connect
to the db
query_rewrite_enabled         = TRUE      # Oracle Retail required for function-
based indexes
session_cached_cursors                  = 900       # Oracle Retail required;
undo_management           = AUTO
undo_retention            = 1800      # Currently set for 30 minutes; set to avg
length of transactions in sec
undo_tablespace           = undo_ts
user_dump_dest            = <admin_path>/udump
utl_file_dir                            = <utl_file_path>
workarea_size_policy                    = auto          # Should be set to auto
when pga_aggregate_target is set
# ***  Archive Logging, set if needed  ***
#log_archive_dest_1                     = 'location=<admin_path>/arch/'
#log_archive_format               = SIDarch_%r_%s_%t.log
#log_buffer           = 10485760    # Set to (512K or 128K)*CPUs
#log_checkpoint_interval                = 51200     # Default:0 - unlimited
#log_checkpoint_timeout                 = 7200      # Default:1800 seconds
```

B

Appendix: Tablespace Creation Scripts

Note: If you have created tablespaces during the implementation or upgrade of the Retail Merchandising System 14, you no longer require to create another set of tablespaces for the Warehouse Management System as they use the same ones.

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:

   ```
   SQL> ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "password";
   ```

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

   ```
   SQL> SELECT * FROM V$ENCRYPTION_WALLET;

   WRL_TYPE  WRL_PARAMETER                           STATUS___
   file      /u00/oracle/admin/(<$ORACLE_SID>)/wallet  OPEN
   ```

In order for the Oracle database to process encrypted data, the wallet needs to be opened after database startup; it stays open until closed or database shutdown:

```
SQL> ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "password";
```

An auto-open wallet does not require a password to be given after database startup. For better security, there is also a local auto-open wallet which only auto-opens on the server it was created on:

```
$ cd /u00/oracle/admin/(<$ORACLE_SID>)/wallet
$ orapki wallet create –wallet . -auto_login_local
$ chmod 600 *
```

Encryption at Tablespace Level

Once the wallet is configured, determine an encryption algorithm to be used 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.

C

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.

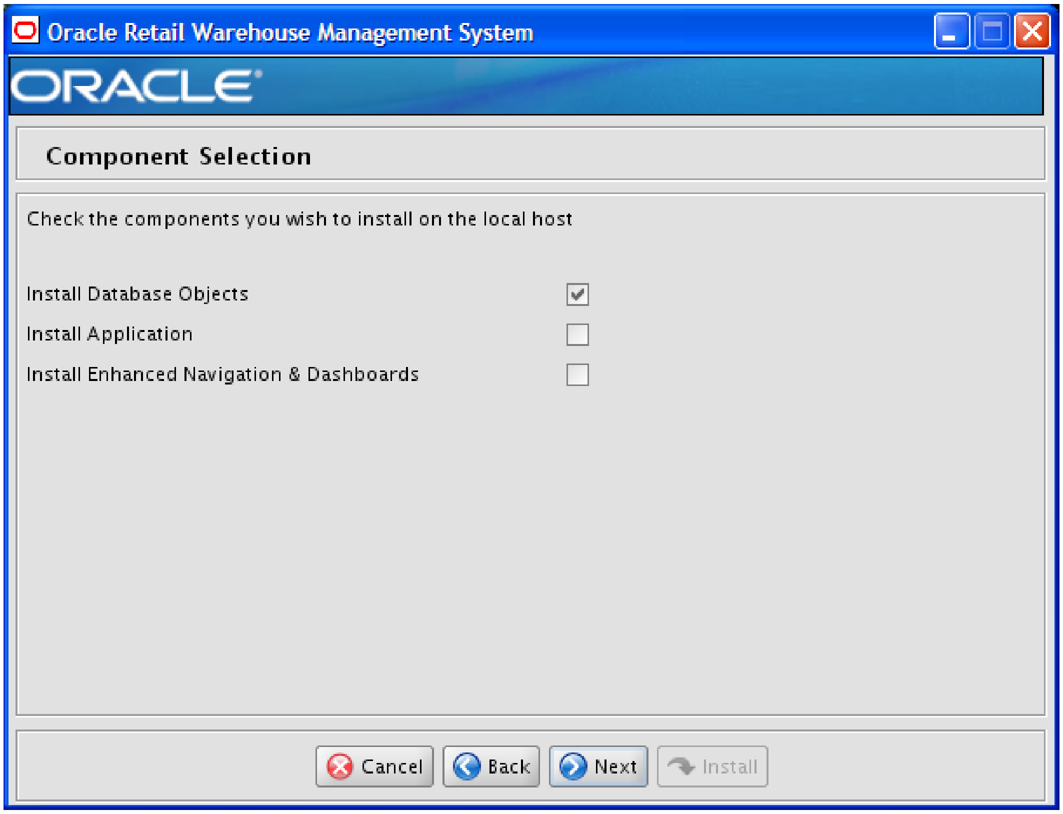
```
####################################################################
# File: tnsnames.ora
# Desc: Net Services configuration file.
# Note: Change these values: <service_name>, <oracle_sid>, <hostname>,
#       <global_name>
####################################################################
<service_name>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

<service_name>.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = <hostname>)(Port = 1521)))
    (CONNECT_DATA = (SID = <oracle_sid>) (GLOBAL_NAME = <global_name>)))

Example:
prod_db1 =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = server_01)(Port = 1521)))
    (CONNECT_DATA = (SID = prod_db1) (GLOBAL_NAME = prod_db1.world)))

prod_db1.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)(host = server_01)(Port = 1521)))
    (CONNECT_DATA = (SID = prod_db1) (GLOBAL_NAME = prod_db1.world)))
```

D

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

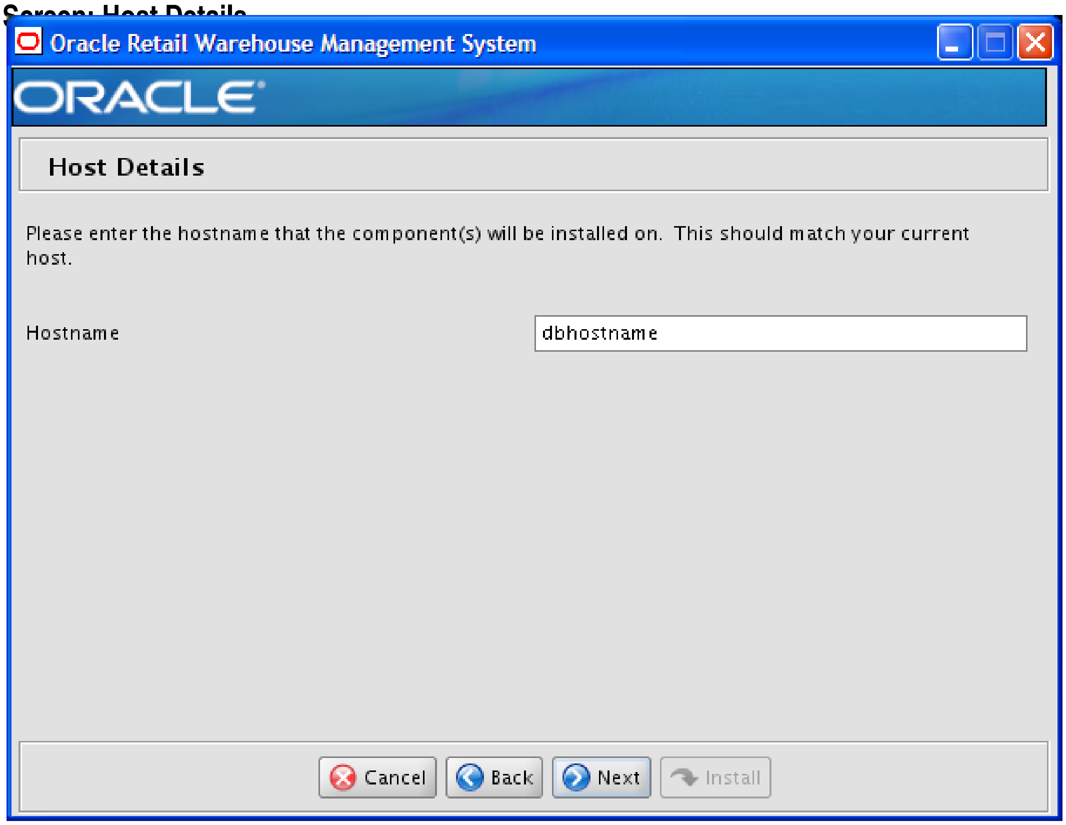


Check the components you wish to install on the local host

Install Database Objects ☑
Install Application □
Install Enhanced Navigation & Dashboards □

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



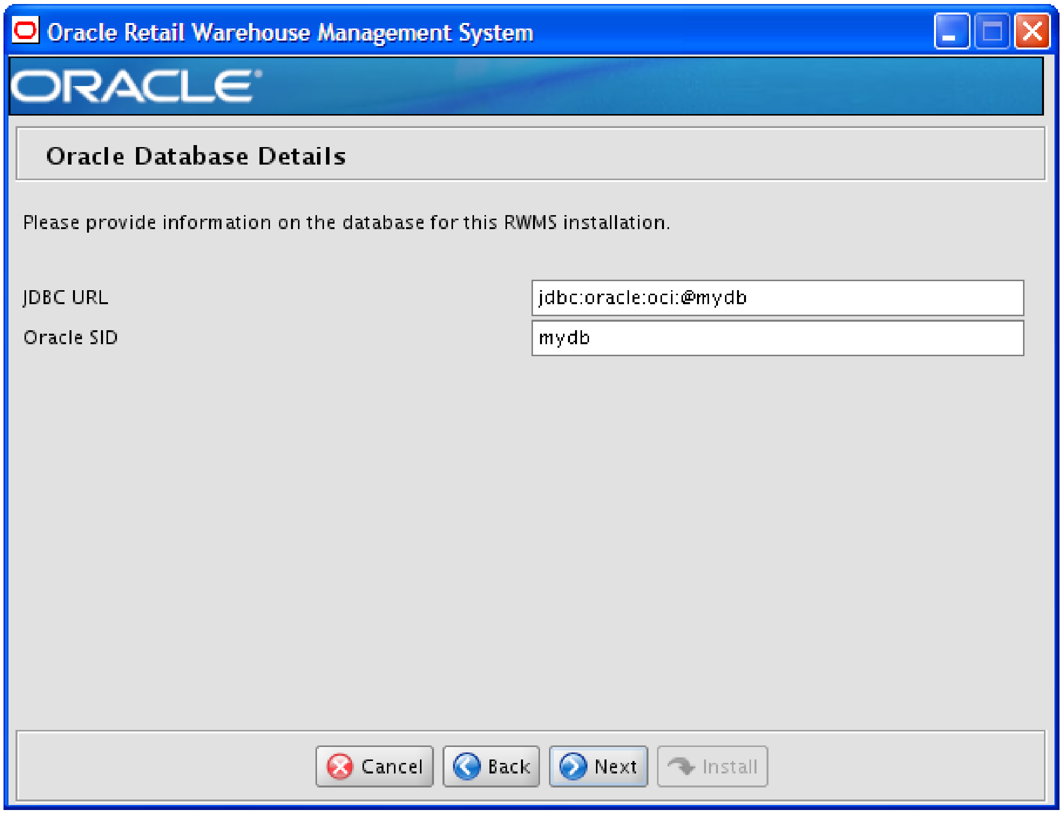
Host Details

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

Hostname: dbhostname

Field Title	Hostname
Field Description	Provide the hostname of the Oracle Database Server
Examples	dbhostname

Screen: Oracle Database Details



Oracle Database Details

Please provide information on the database for this RWMS installation.

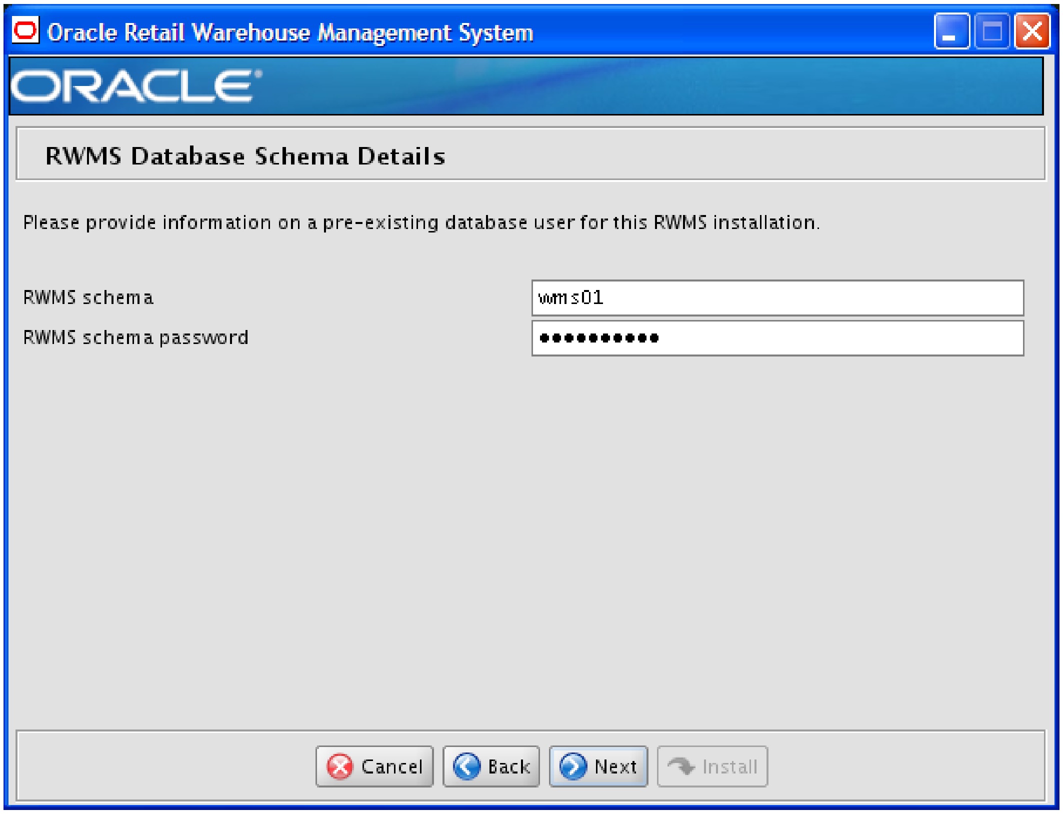
JDBC URL: jdbc:oracle:oci:@mydb

Oracle SID: mydb

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	Standard Thin Connection: jdbc:oracle:thin:@dbhostname:1521:mydb Standard OCI Connection: jdbc:oracle:thin:@mydb 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)))

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



RWMS Database Schema Details

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

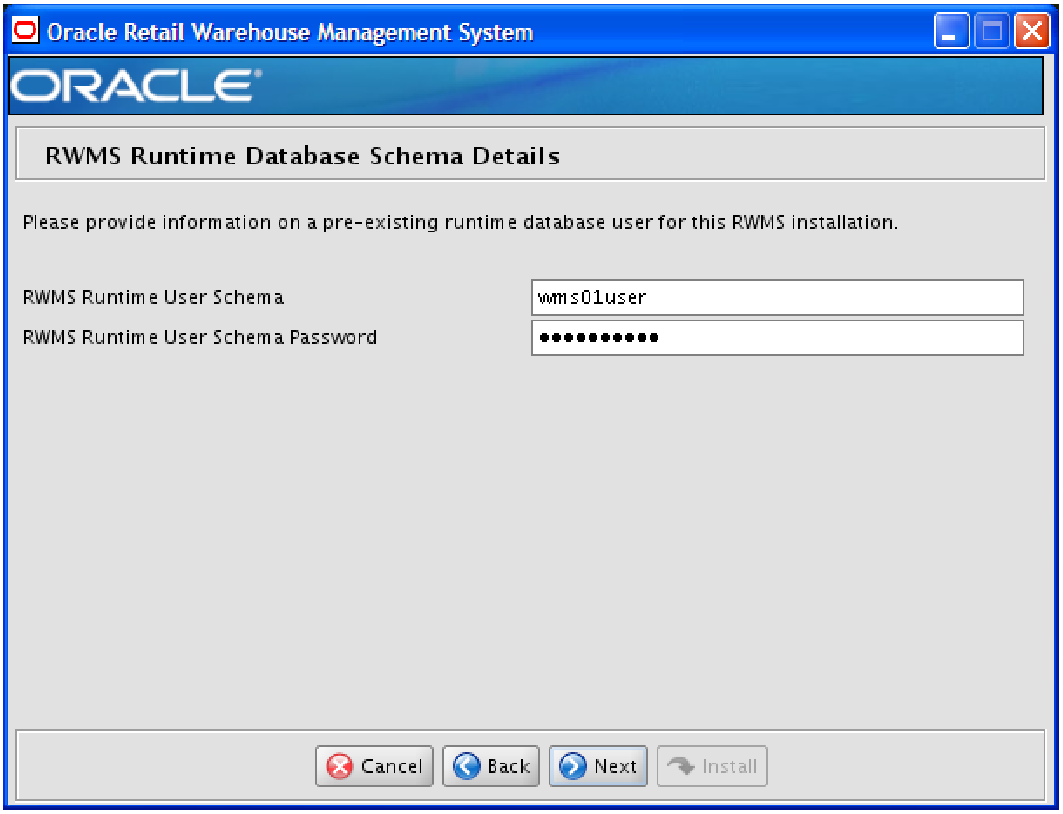
RWMS schema: wms01

RWMS schema password: ●●●●●●●●●●●

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



RWMS Runtime Database Schema Details

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

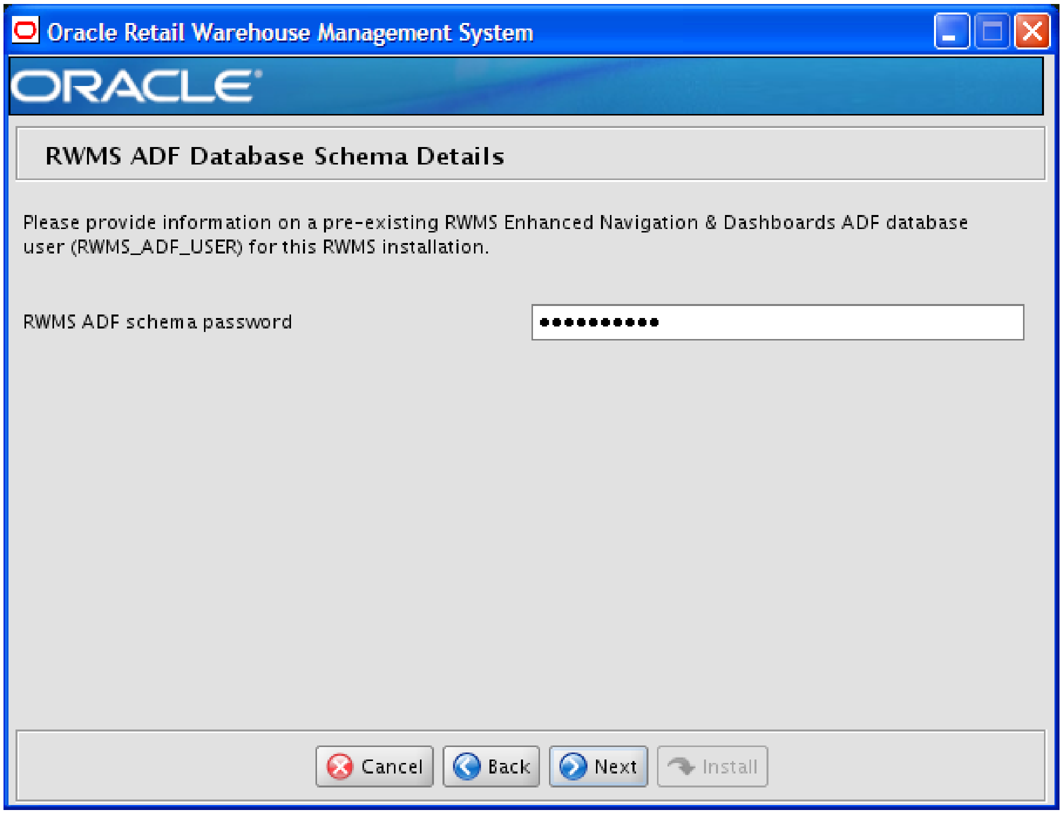
RWMS Runtime User Schema: wms01user

RWMS Runtime User Schema Password: ●●●●●●●●●●●

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



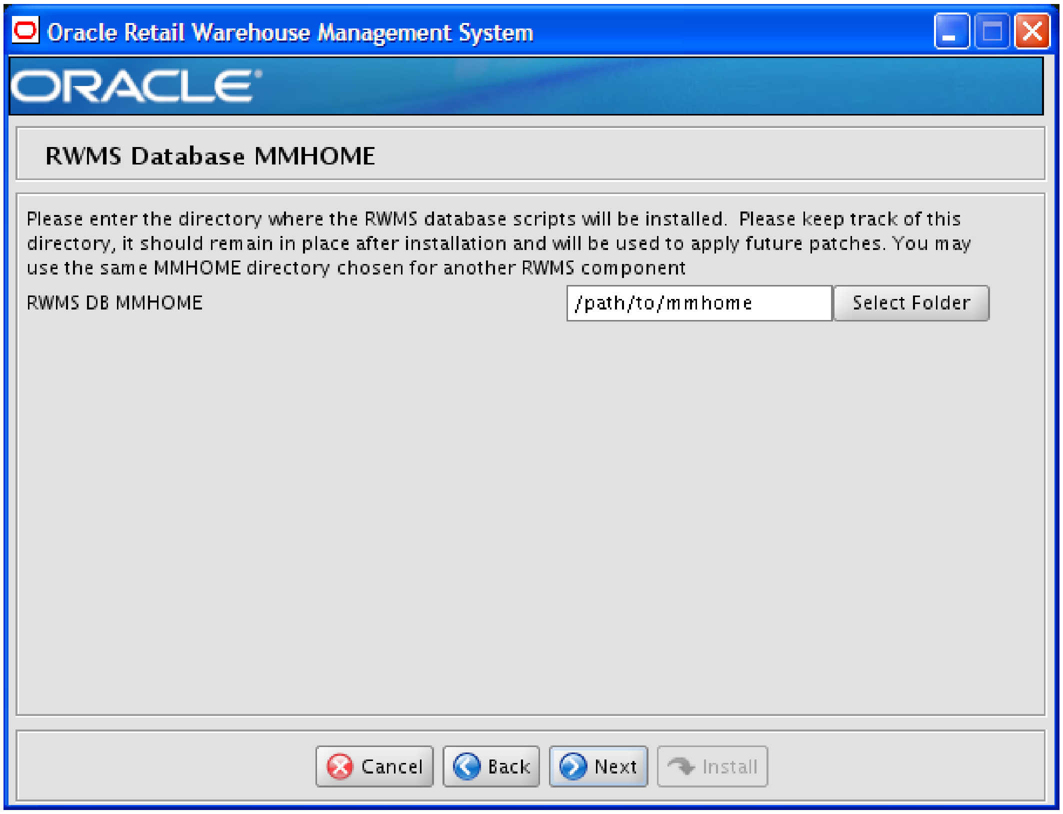
RWMS ADF Database Schema Details

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 ●●●●●●●●●●●

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 MMHOME

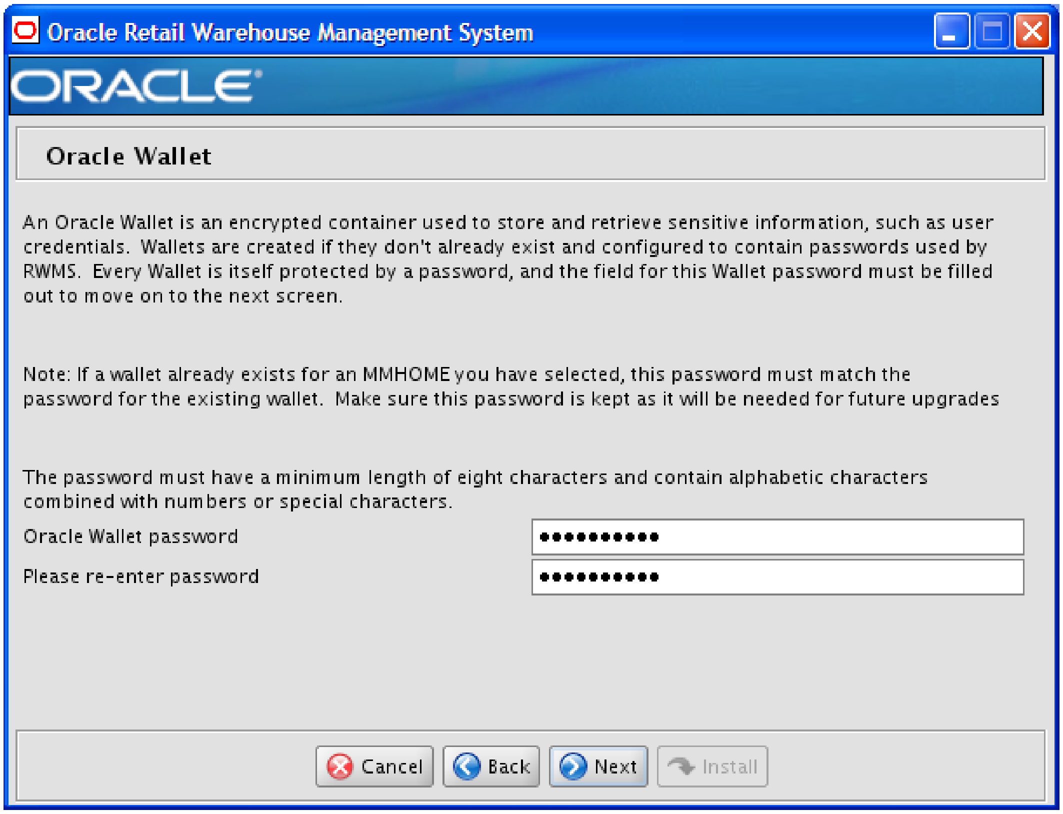


Please enter the directory where the RWMS database scripts will be installed. Please keep track of this directory, it should remain in place after installation and will be used to apply future patches. You may use the same MMHOME directory chosen for another RWMS component

RWMS DB MMHOME /path/to/mmhome

Field Title	RWMS DB MMHOME
Field Description	The location where the RWMS database scripts will be installed along with the ORPATCH utility.
Example	/path/to/mmhome
Note	If you have selected an existing MMHOME, 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 an MMHOME 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: ●●●●●●●●●●●

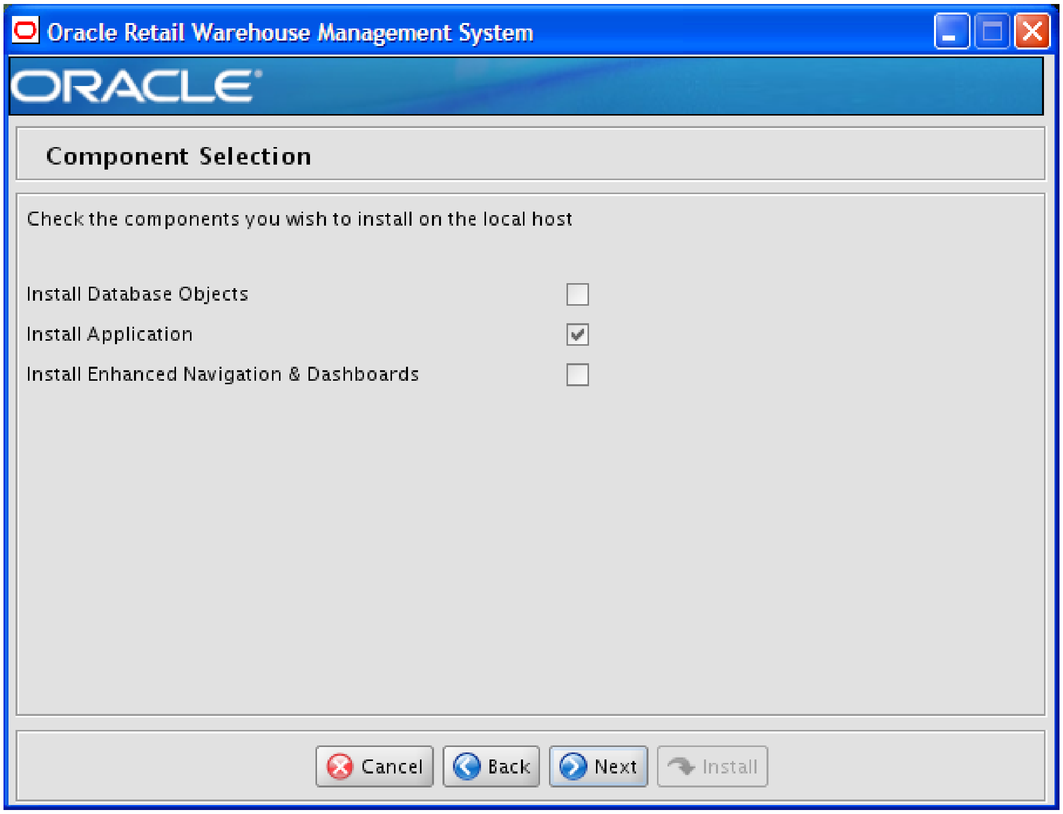
Please re-enter password: ●●●●●●●●●●●

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 MMHOME in the previous screens, you will need to enter the password that was used for the wallet in that MMHOME
Note	Make sure this password is kept as it will be needed for future upgrades.

E

Appendix: RWMS Application(forms) Installation Screens

Screen: Component Selection



Check the components you wish to install on the local host

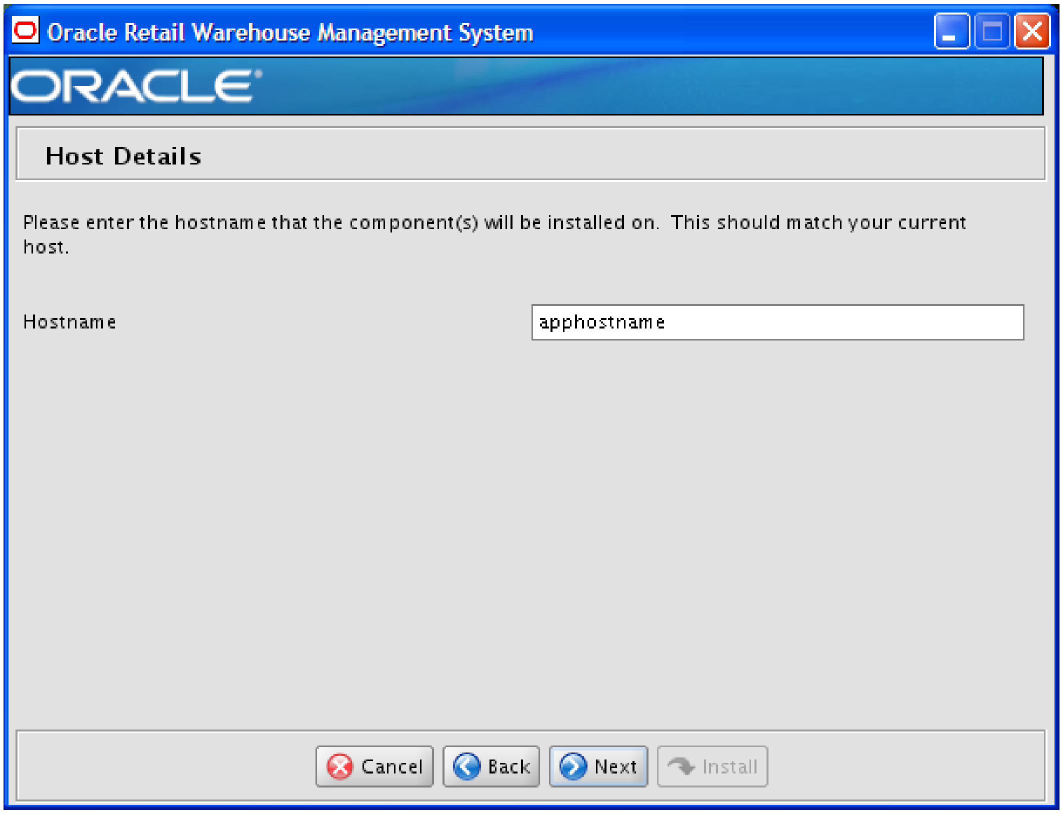
Install Database Objects □

Install Application ☑

Install Enhanced Navigation & Dashboards □

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

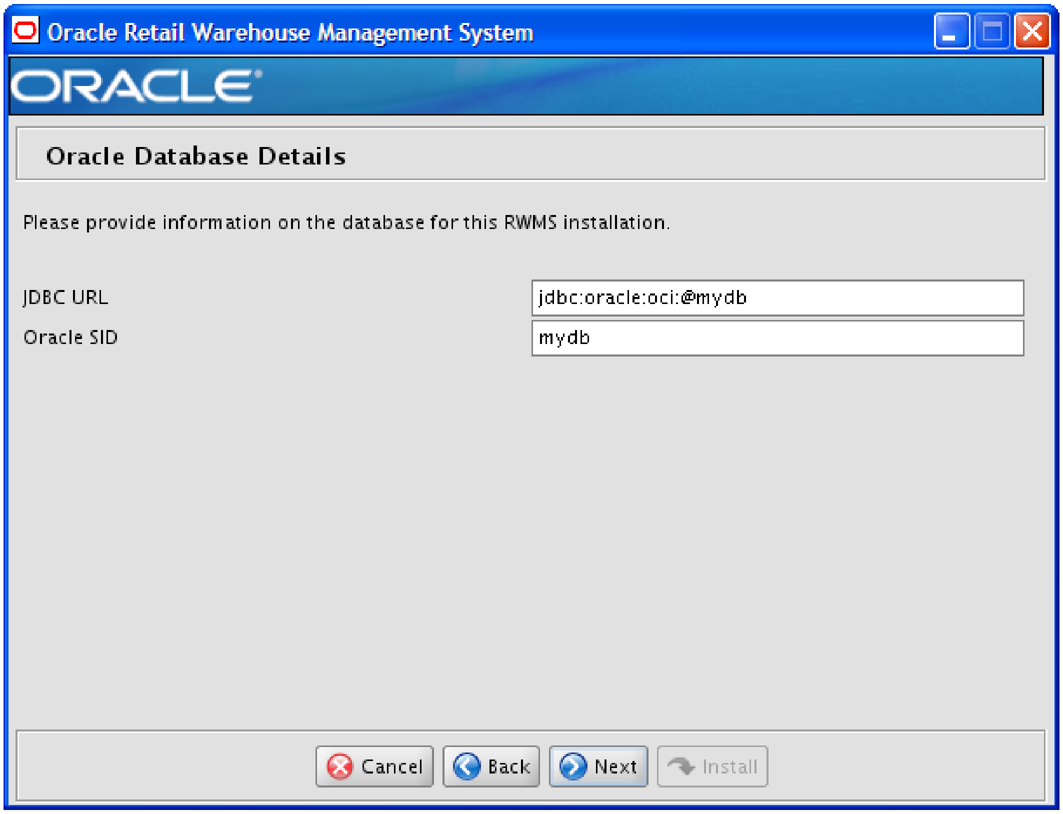


Please enter the hostname that the component(s) will be installed on. This should match your current host.

Hostname: apphostname

Field Title	Hostname
Field Description	Provide the hostname of where the application needs to be installed
Examples	apphostname

Screen: Oracle Database Details



Oracle Database Details

Please provide information on the database for this RWMS installation.

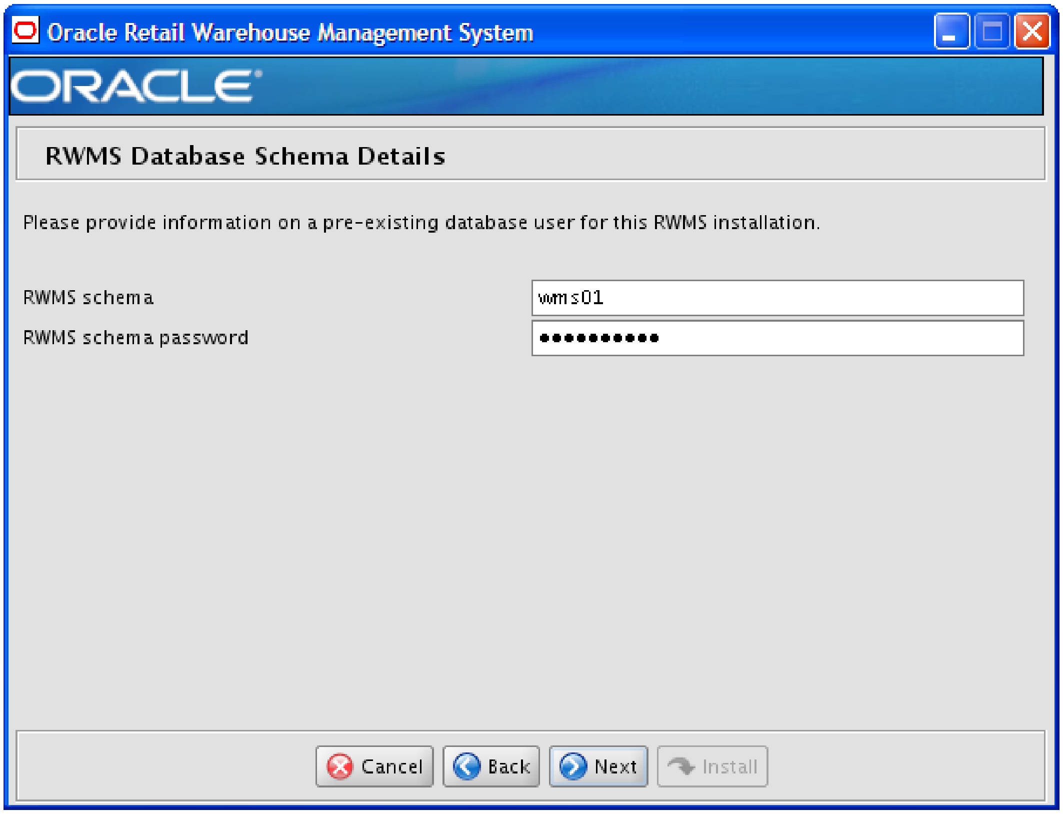
JDBC URL jdbc:oracle:oci:@mydb

Oracle SID mydb

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	Standard Thin Connection: jdbc:oracle:thin:@dbhostname:1521:mydb Standard OCI Connection: jdbc:oracle:thin:@mydb 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)))

Field Title	Oracle SID
Field Description	Oracle system identifier for the database that the RWMS installation will connect to.
Examples	mydb

Screen: Database Schema Details



RWMS Database Schema Details

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

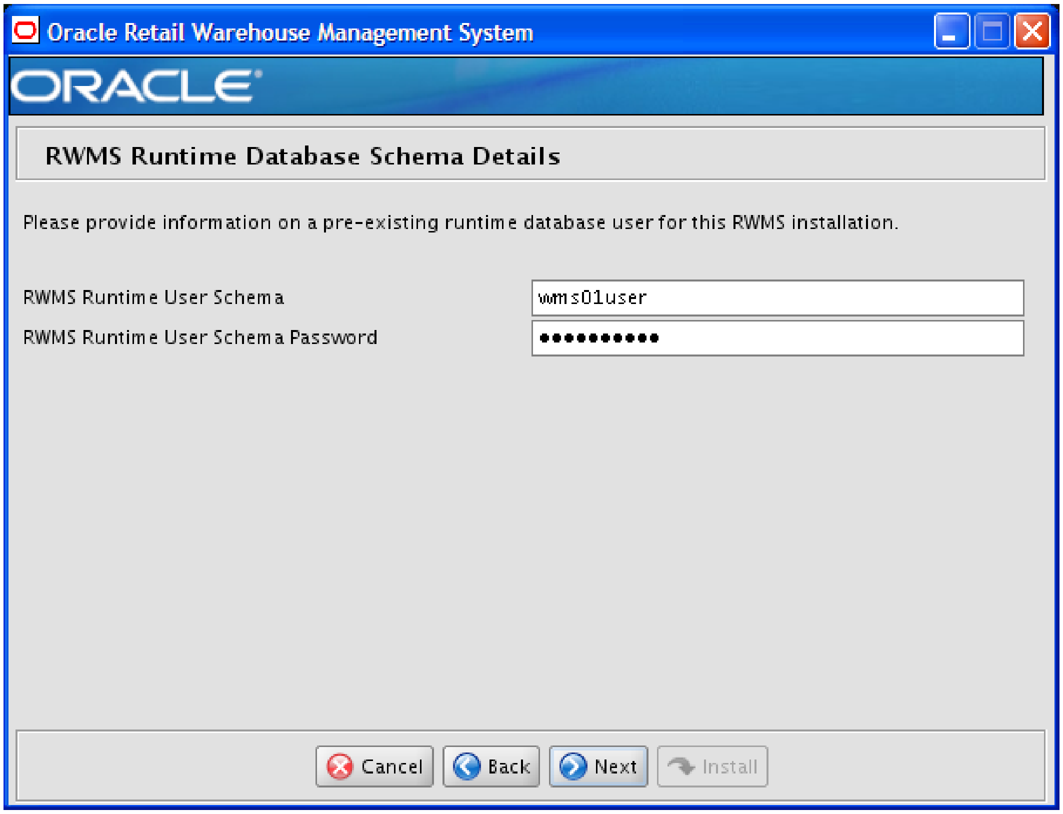
RWMS schema: wms01

RWMS schema password: ●●●●●●●●●●●

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: Runtime Database Schema Details



RWMS Runtime Database Schema Details

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

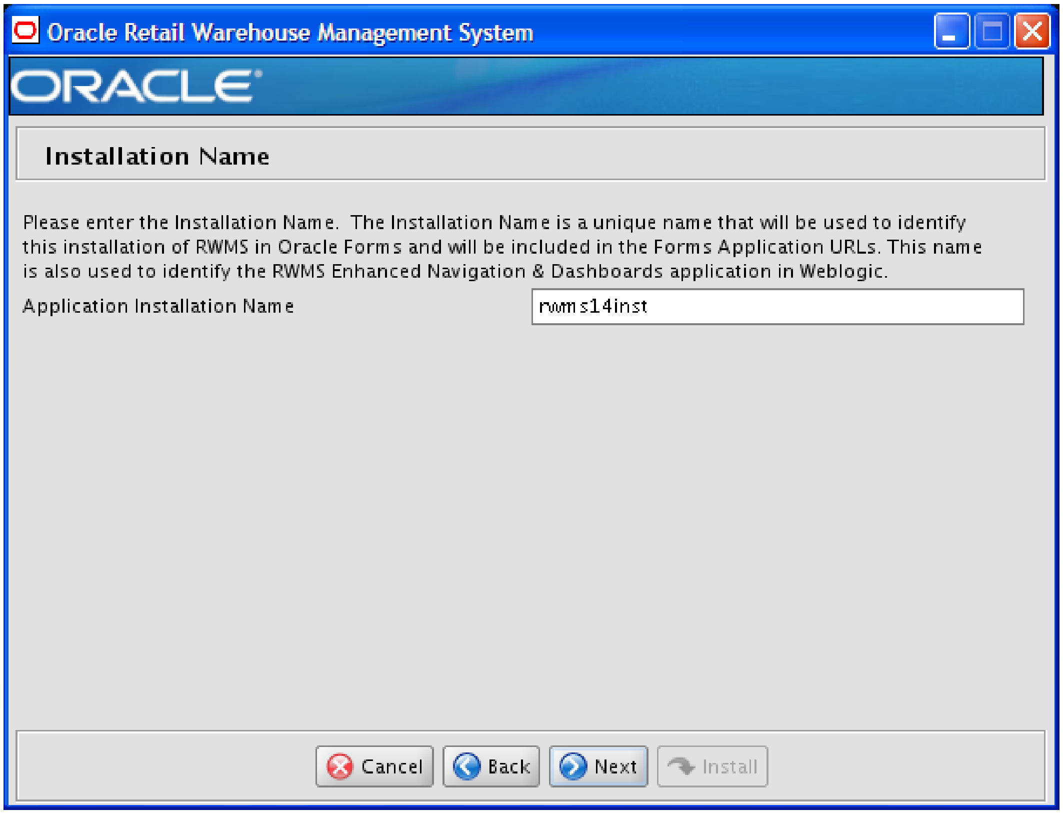
RWMS Runtime User Schema: wms01user

RWMS Runtime User Schema Password: ●●●●●●●●●●●

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



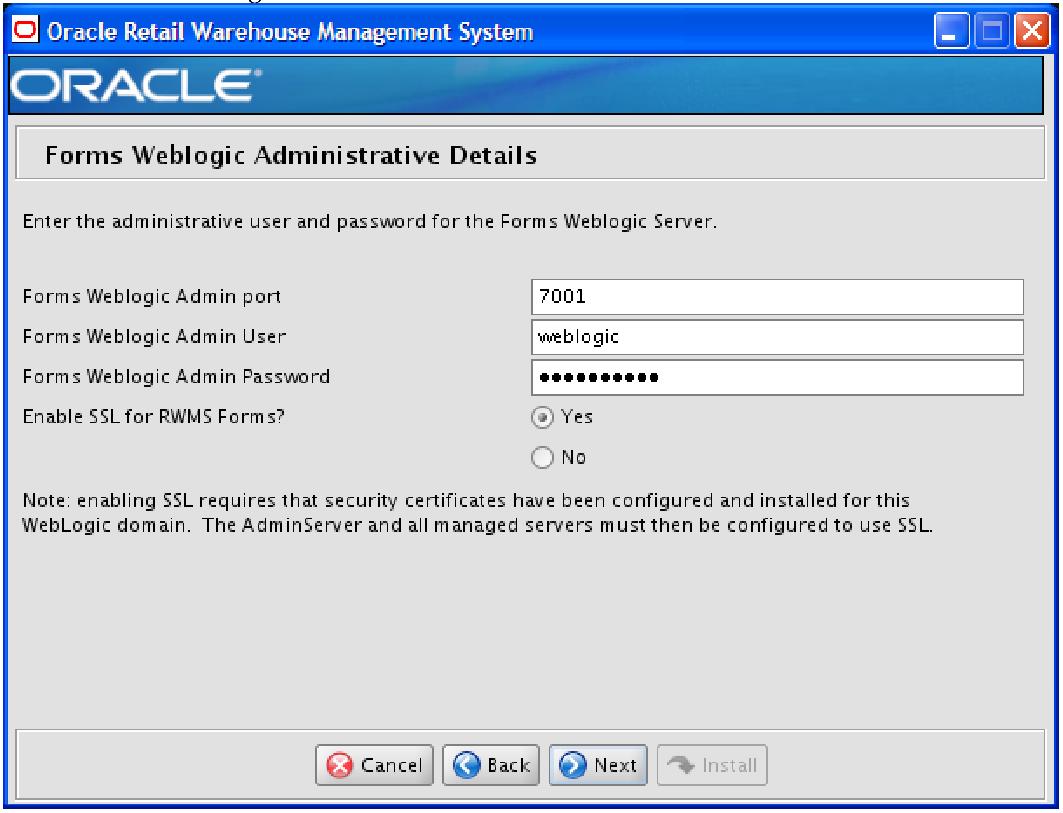
Installation Name

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: rwms14inst

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	rwms14inst
Note	The value entered in this field must be identical for the Application (Forms) and Enhanced Navigation & Dashboards installations.

Screen: Forms Weblogic Administrative Details



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: ●●●●●●●●●●●

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.

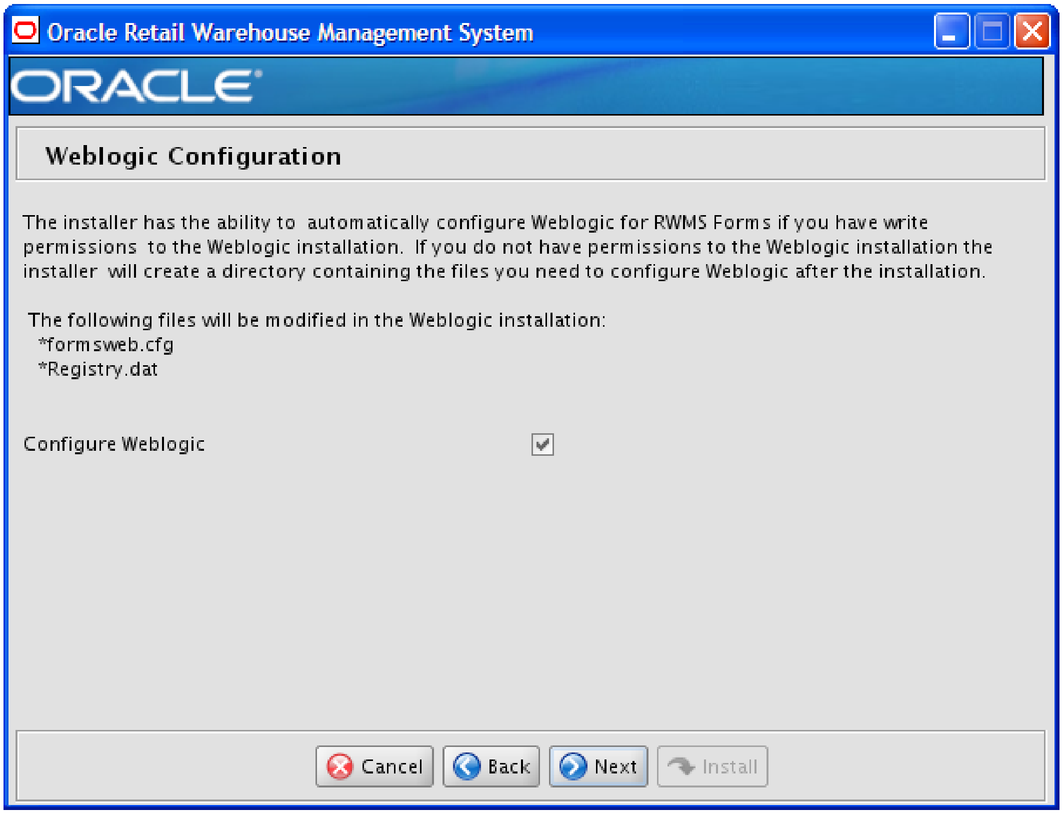
Field Title	Forms WebLogic Admin Port
Field Description	Listen port for the Forms WebLogic Admin server
Example	7001

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



Weblogic Configuration

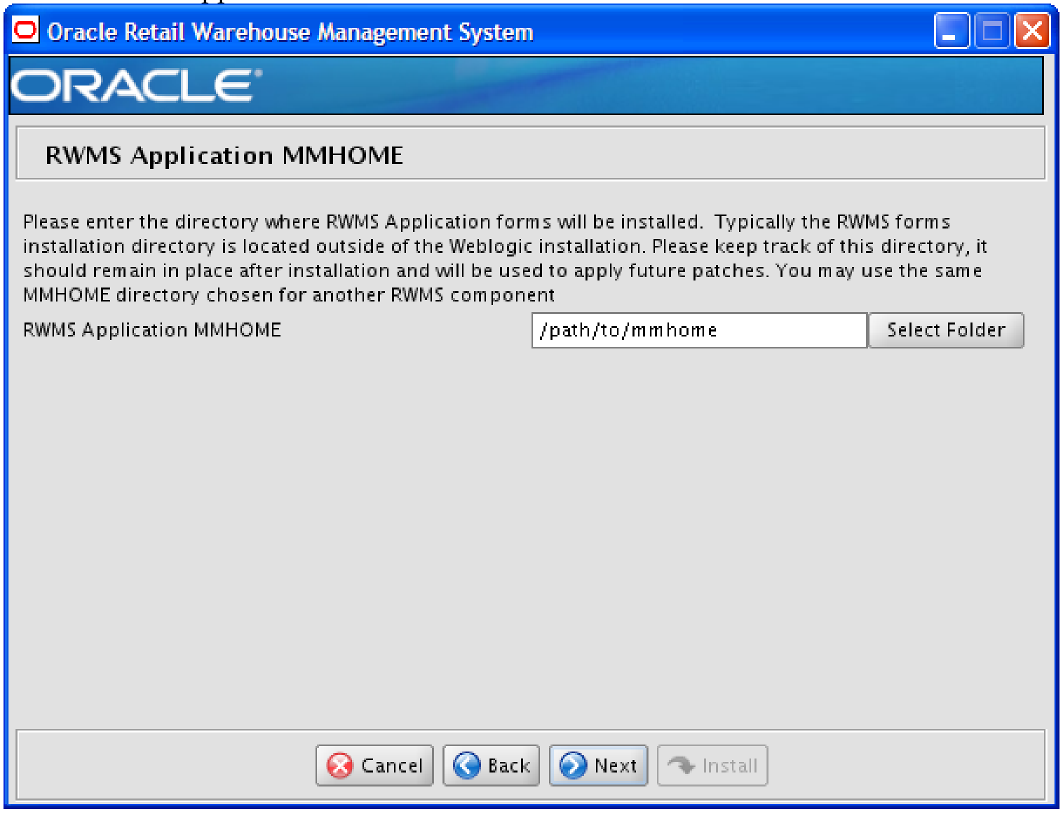
The installer has the ability to automatically configure Weblogic for RWMS Forms if you have write permissions to the Weblogic installation. If you do not have permissions to the Weblogic installation the installer will create a directory containing the files you need to configure Weblogic after the installation.

The following files will be modified in the Weblogic installation:
*formsweb.cfg
*Registry.dat

Configure Weblogic ☑

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 filess will be generated by the installer, but should be applied to WebLogic manually.

Screen: RWMS Application MMHOME



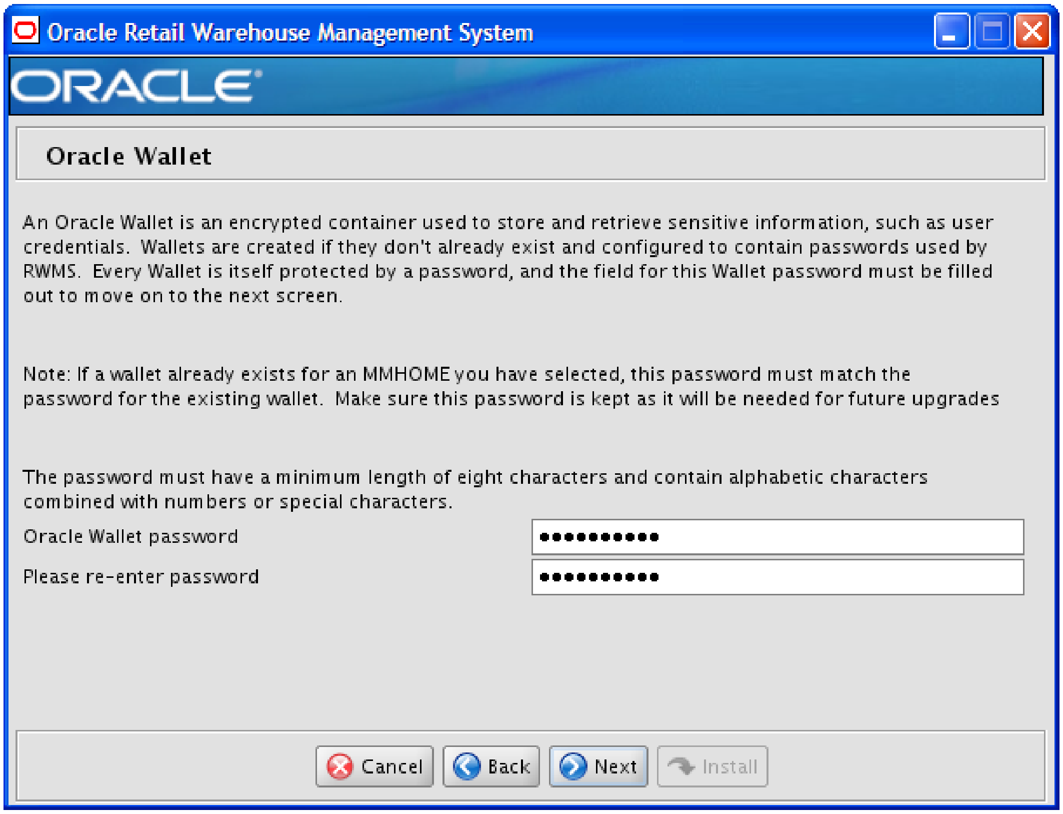
RWMS Application MMHOME

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 to apply future patches. You may use the same MMHOME directory chosen for another RWMS component

RWMS Application MMHOME: /path/to/mmhome

Field Title	RWMS Application MMHOME
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/mmhome
Note	If you have selected an existing MMHOME, 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 an MMHOME 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 ●●●●●●●●●●●

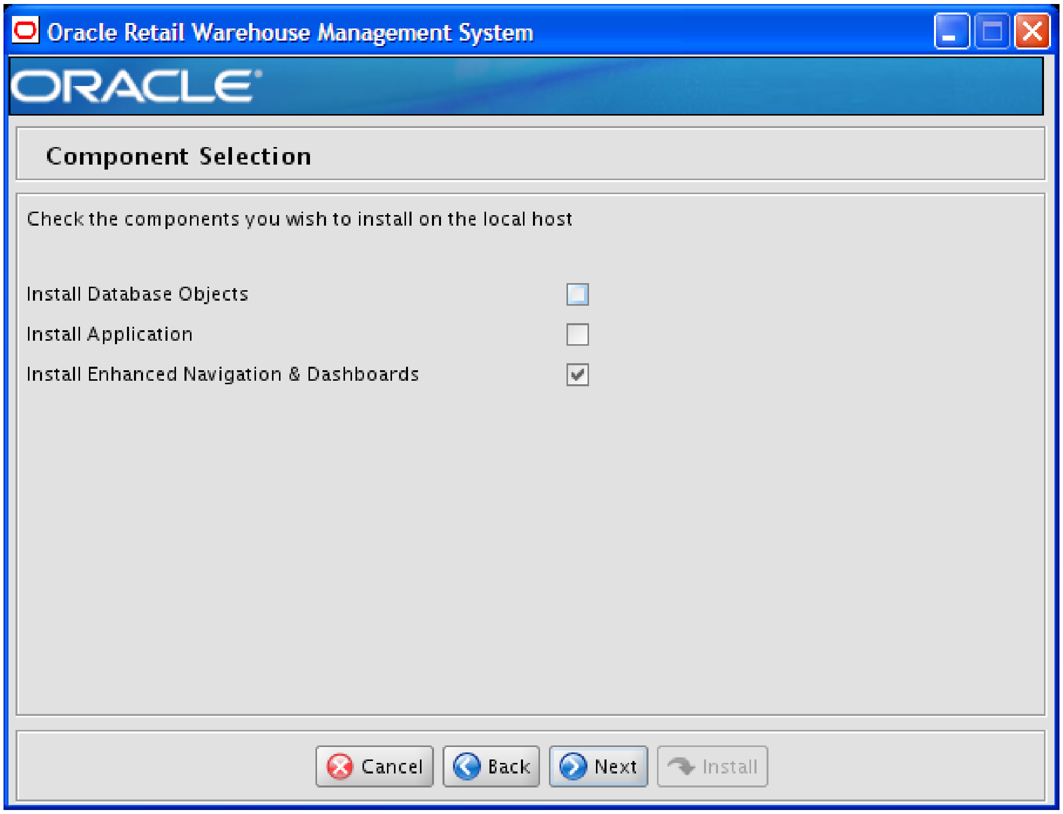
Please re-enter password ●●●●●●●●●●●

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 MMHOME in the previous screens, you will need to enter the password that was used for the wallet in that MMHOME
Note	Make sure this password is kept as it will be needed for future upgrades.

F

Appendix: RWMS Enhanced Navigation and Dashboards Installation Screens

Screen: Component Selection



Oracle Retail Warehouse Management System

Component Selection

Check the components you wish to install on the local host

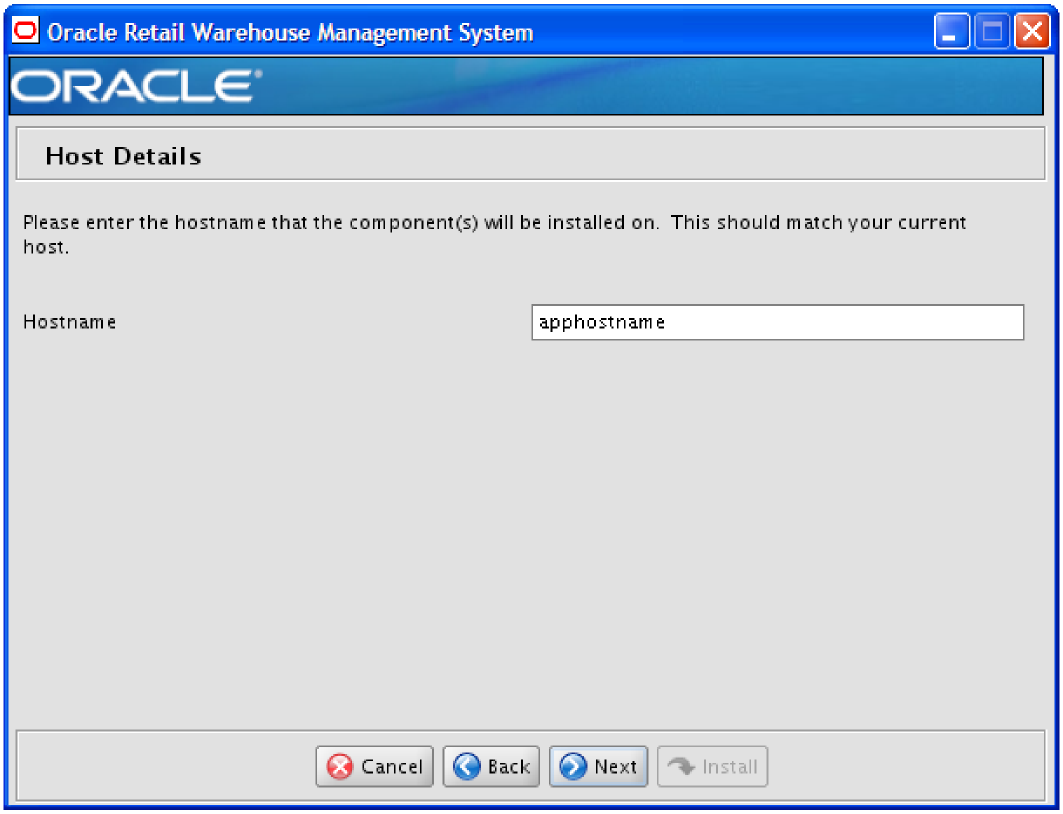
Install Database Objects □

Install Application □

Install Enhanced Navigation & Dashboards ☑

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



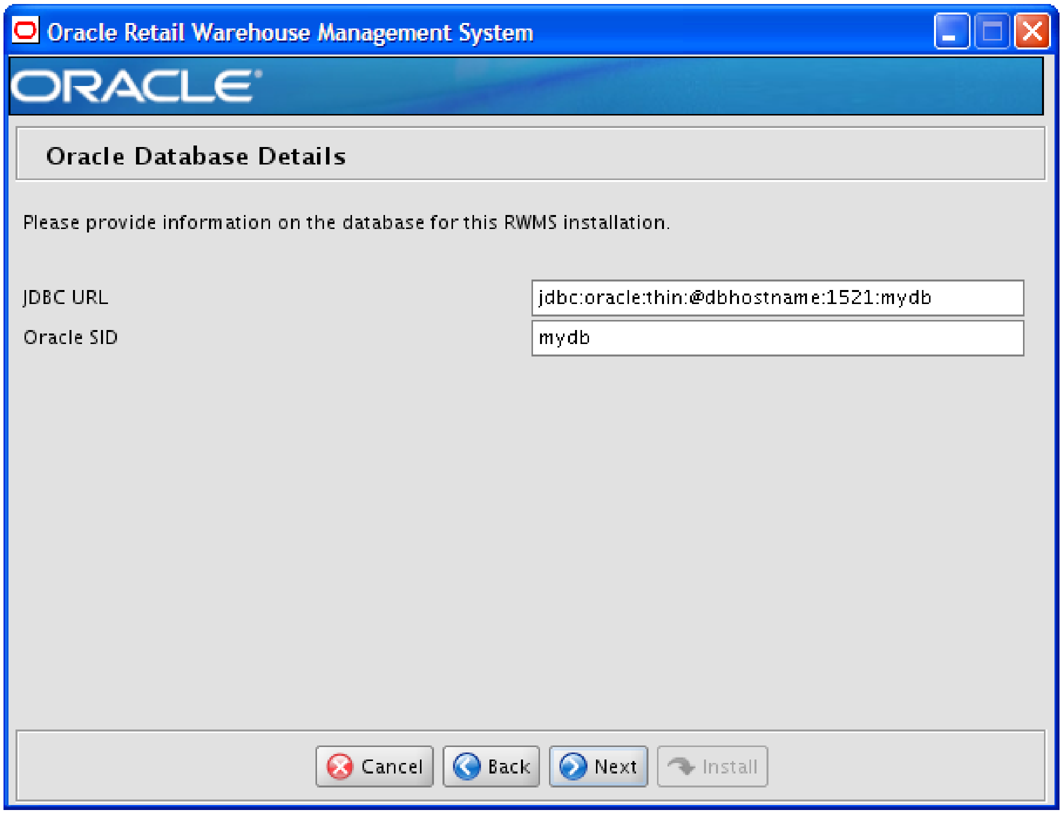
Host Details

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

Hostname: apphostname

Field Title	Hostname
Field Description	Provide the hostname of where the application needs to be installed
Examples	apphostname

Screen: Database Details



Please provide information on the database for this RWMS installation.

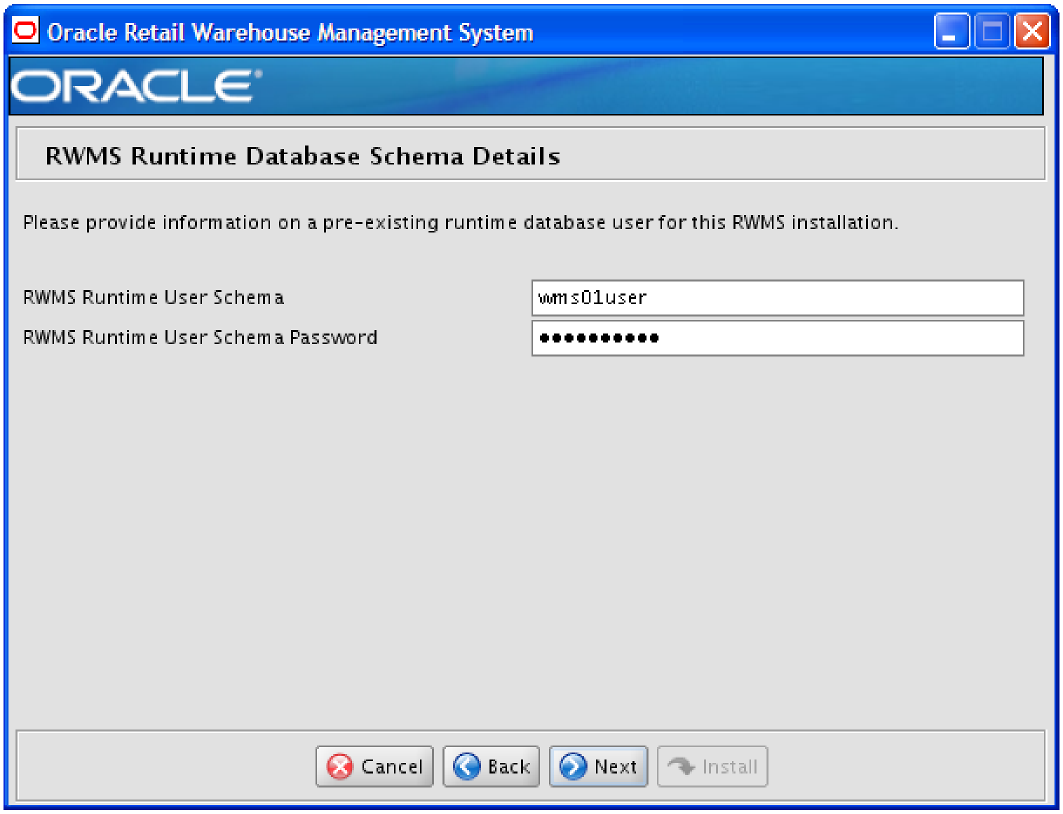
JDBC URL: jdbc:oracle:thin:@dbhostname:1521:mydb

Oracle SID: mydb

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	Standard Thin Connection: jdbc:oracle:thin:@dbhostname:1521:mydb Standard OCI Connection: jdbc:oracle:thin:@mydb 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)))

Field Title	Oracle SID
Field Description	Oracle system identifier for the database that the RWMS installation will connect to.
Examples	mydb

Screen: Runtime Database Schema Details



RWMS Runtime Database Schema Details

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

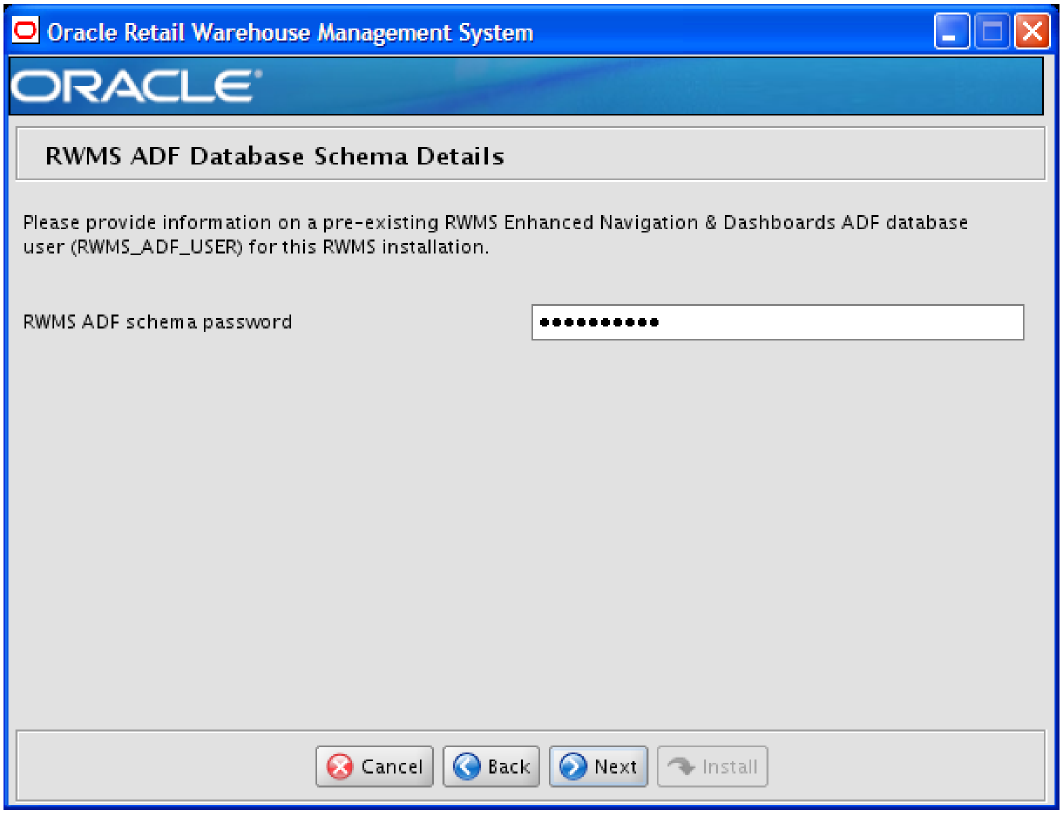
RWMS Runtime User Schema: wms01user

RWMS Runtime User Schema Password: ●●●●●●●●●●●

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: ADF Database Schema Details



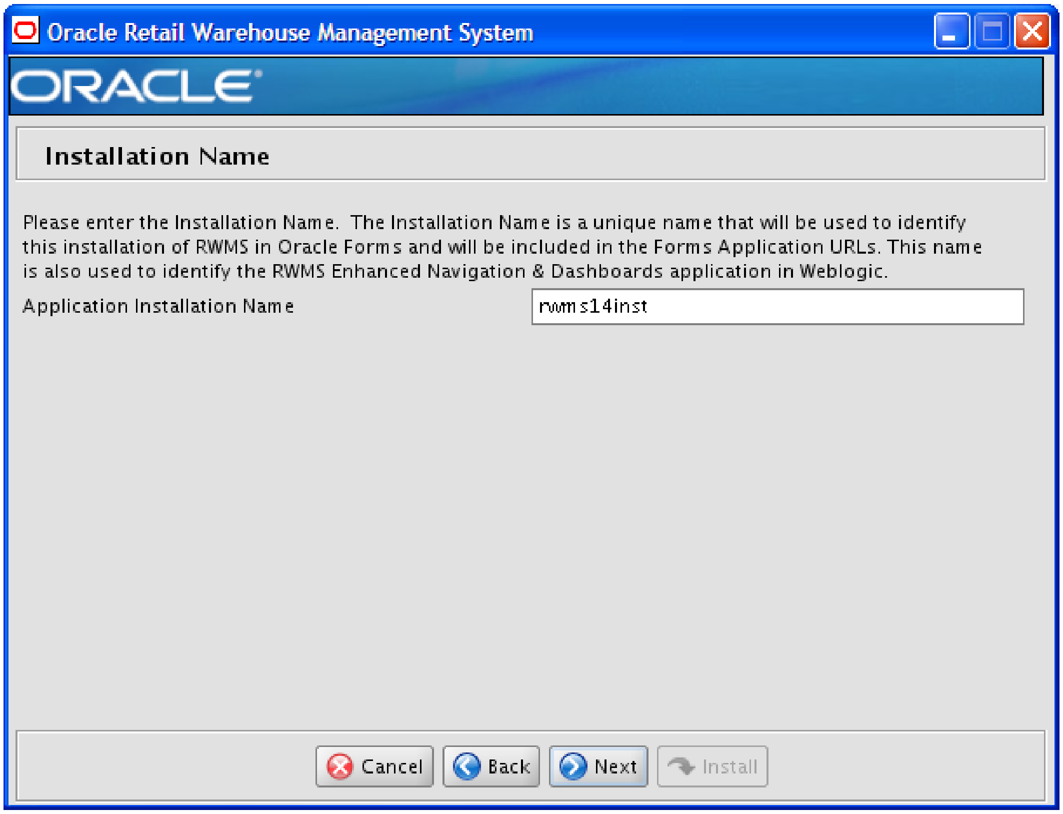
RWMS ADF Database Schema Details

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 ●●●●●●●●●●●

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



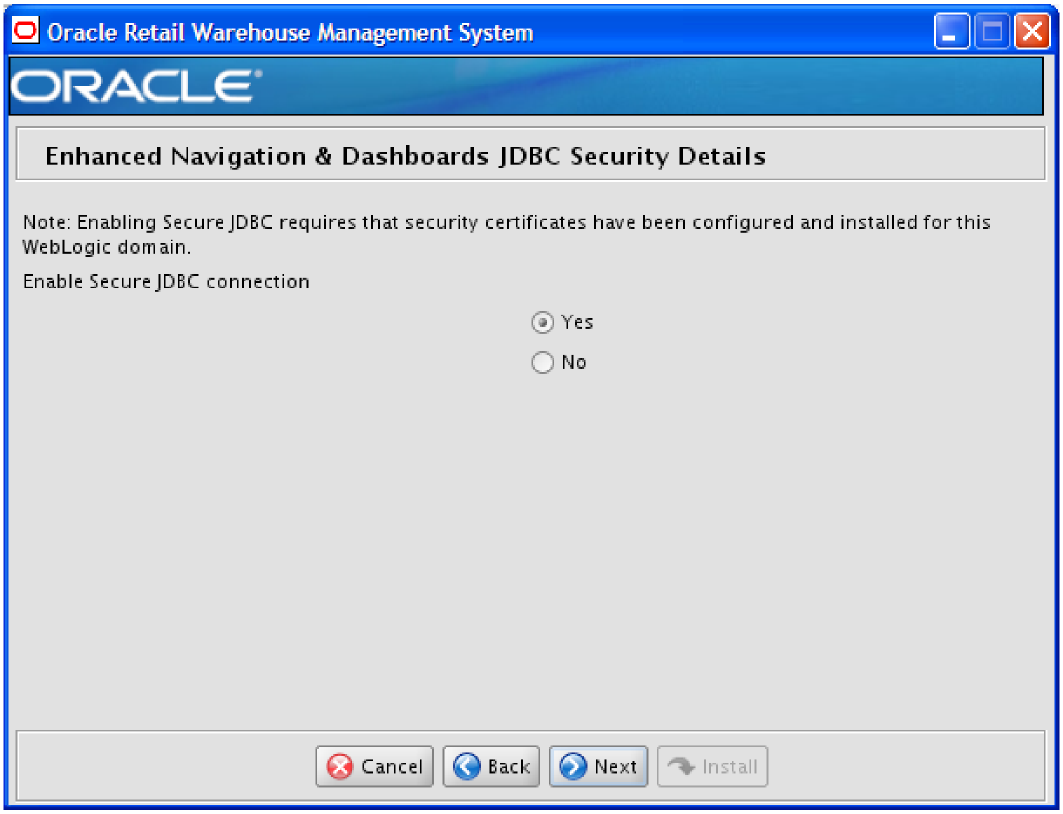
Installation Name

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: rwms14inst

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	rwms14inst
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



Enhanced Navigation & Dashboards JDBC Security Details

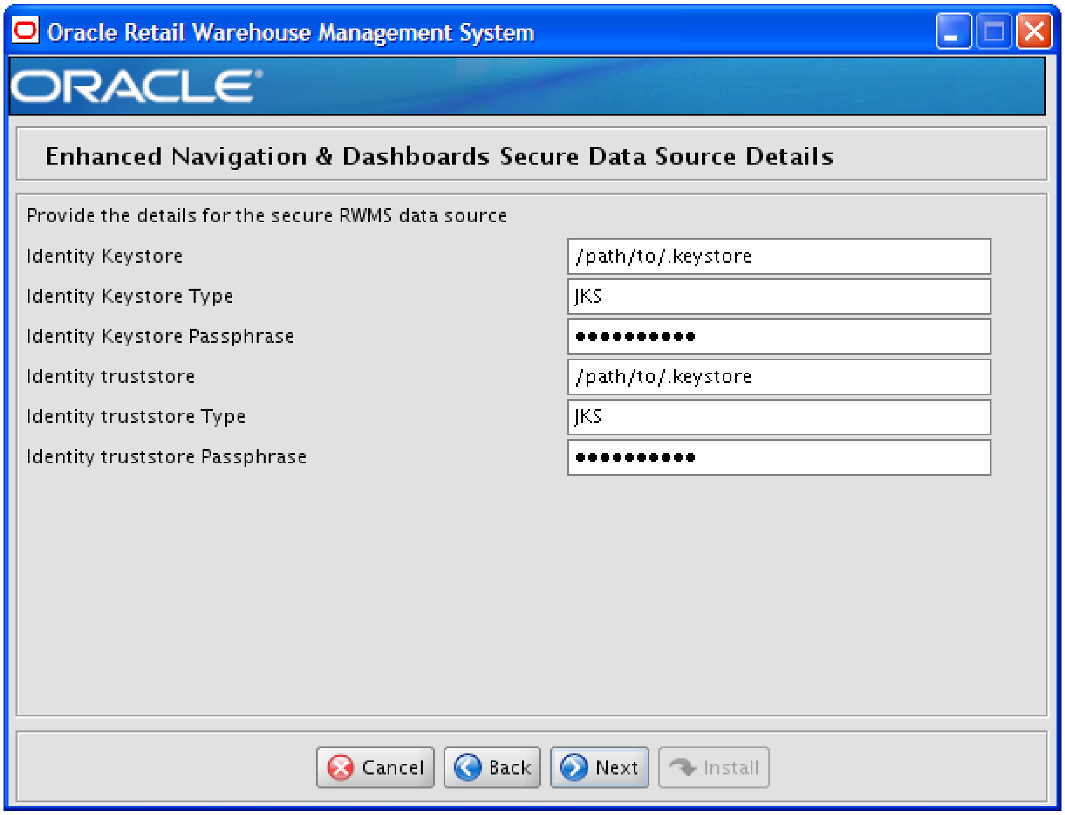
Note: Enabling Secure JDBC requires that security certificates have been configured and installed for this WebLogic domain.

Enable Secure JDBC connection

● Yes
○ No

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



Enhanced Navigation & Dashboards Secure Data Source Details

Provide the details for the secure RWMS data source

Identity Keystore	/path/to/.keystore
Identity Keystore Type	JKS
Identity Keystore Passphrase	●●●●●●●●●●●
Identity truststore	/path/to/.keystore
Identity truststore Type	JKS
Identity truststore Passphrase	●●●●●●●●●●●

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

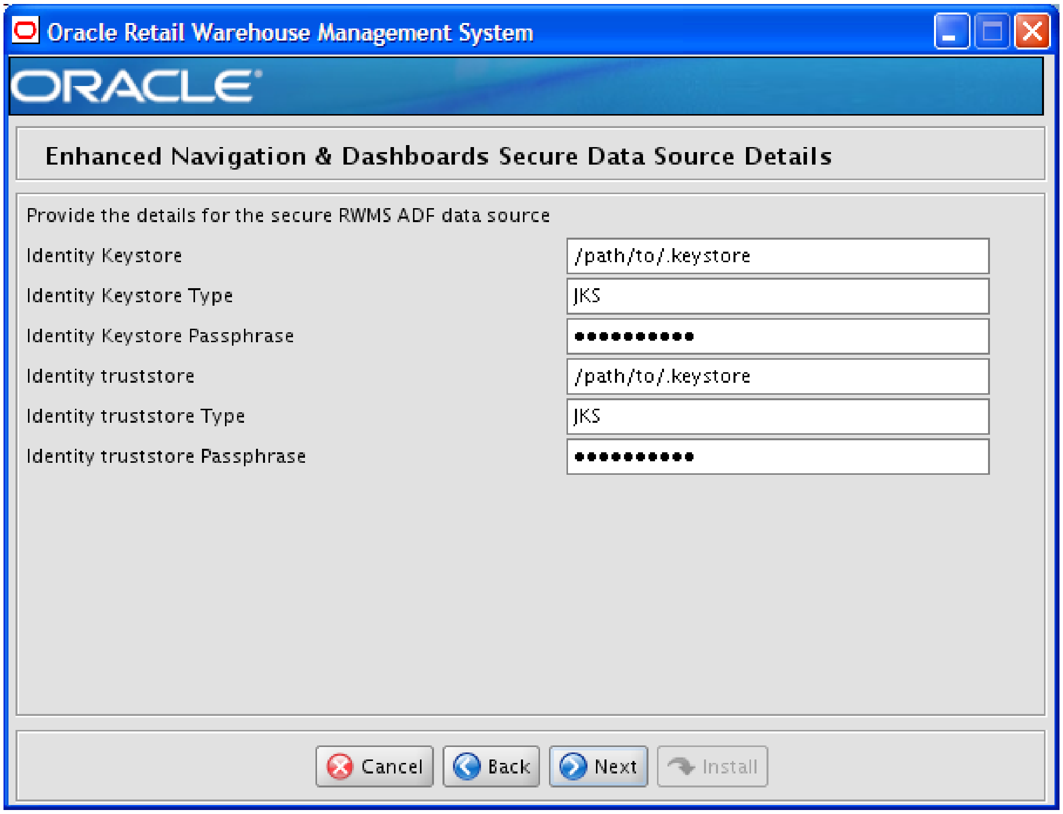
Field Title	Identity Keystore Password
Field Description	Password used to access the identity keystore defined above

Field Title	Identity Trustore
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



Enhanced Navigation & Dashboards Secure Data Source Details

Provide the details for the secure RWMS ADF data source

Identity Keystore	/path/to/.keystore
Identity Keystore Type	JKS
Identity Keystore Passphrase	●●●●●●●●●●●
Identity truststore	/path/to/.keystore
Identity truststore Type	JKS
Identity truststore Passphrase	●●●●●●●●●●●

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

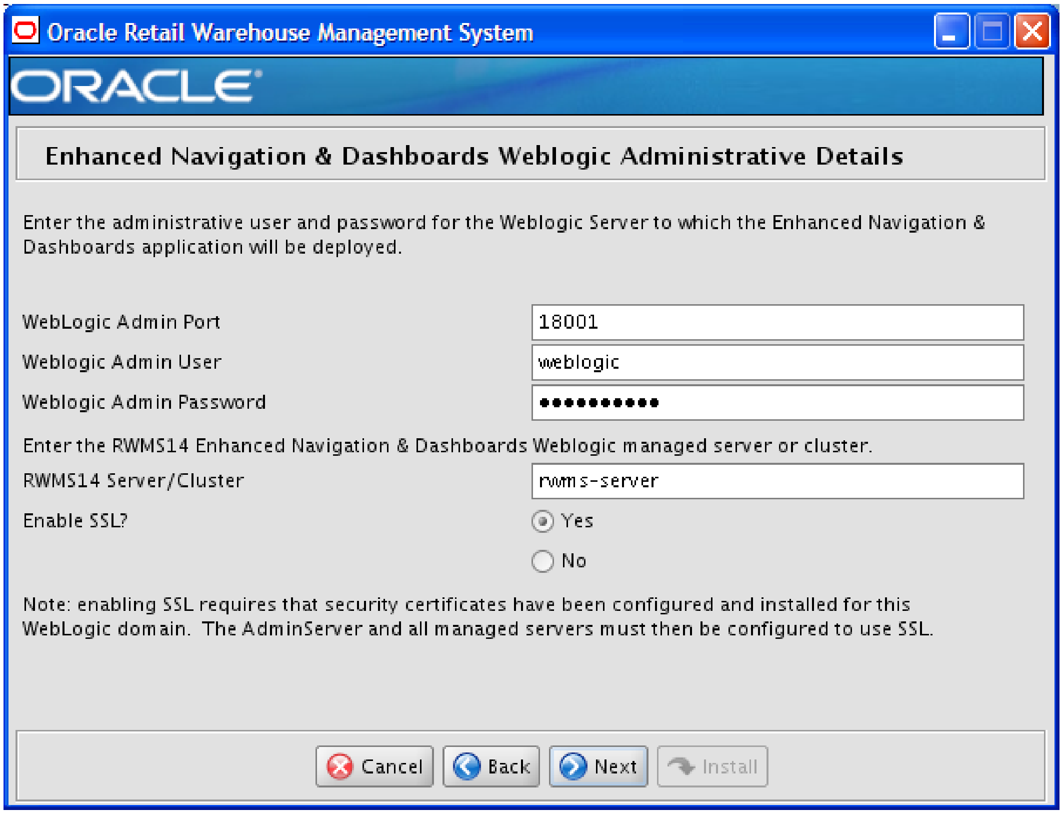
Field Title	Identity Keystore Password
Field Description	Password used to access the identity keystore defined above

Field Title	Identity Trustore
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: ●●●●●●●●●●

Enter the RWMS14 Enhanced Navigation & Dashboards Weblogic managed server or cluster.

RWMS14 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.

Field Title	WebLogic Admin Port
Field Description	Listen port for the Enhanced Navigation & Dashboards WebLogic Admin server
Example	18001

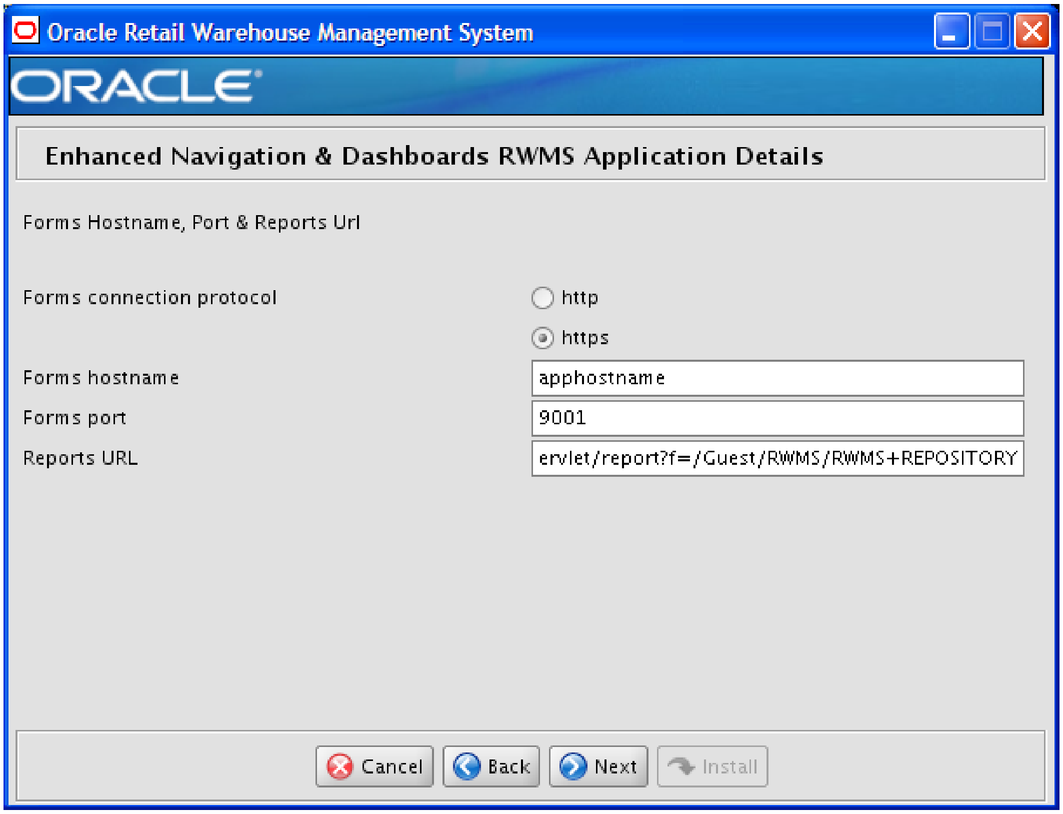
Field Title	WebLogic Admin User
Field Description	Username of the admin user for the WebLogic instance to which the Enhanced Navigation & Dashboards application will be deployed.
Example	weblogic

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: Enhanced Navigation & Dashboards RWMS Application details


Oracle Retail Warehouse Management System
ORACLE
Enhanced Navigation & Dashboards RWMS Application Details
Cancel
Back
Next
Install

Forms Hostname, Port & Reports Url

Forms connection protocol ○ http ● https

Forms hostname: apphostname

Forms port: 9001

Reports URL: ervlet/report?f=/Guest/RWMS/RWMS+REPOSITORY

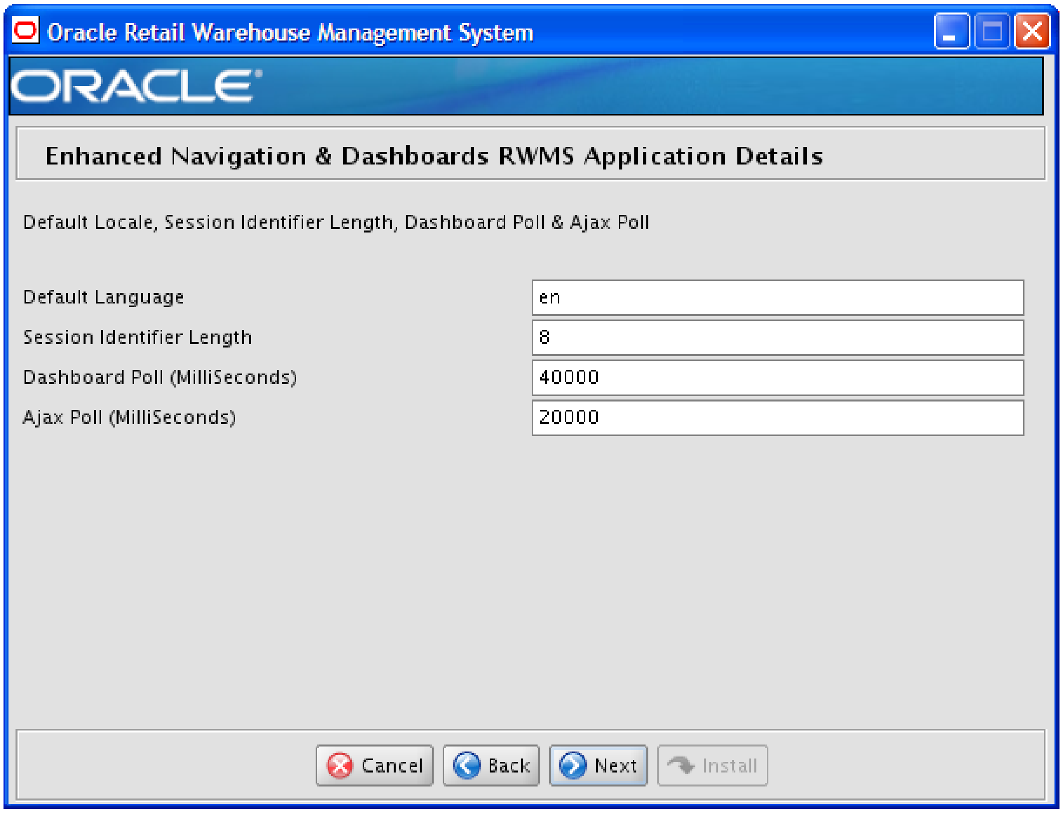
Field Title	Forms connection protocol
Field Description	Select the correct protocol value depending on whether or not RWMS forms was installed using a WebLogic environment configured to use SSL. This should match the selection choosen for the field “Enable SSL for RWMS Forms?” used during the RWMS Forms Application installation. If “Yes” was choosen, 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

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	Provide the URL to access RWMS Reports.
Examples	https://bihostname:19001/xmlpserver/servlet/report?f=/Guest/RWMS/RWMS+REPOSITORY

Screen: Enhanced Navigation & Dashboards RWMS Application Details



Enhanced Navigation & Dashboards RWMS Application Details

Default Locale, Session Identifier Length, Dashboard Poll & Ajax Poll

Default Language: en

Session Identifier Length: 8

Dashboard Poll (MilliSeconds): 40000

Ajax Poll (MilliSeconds): 20000

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

G

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.

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:

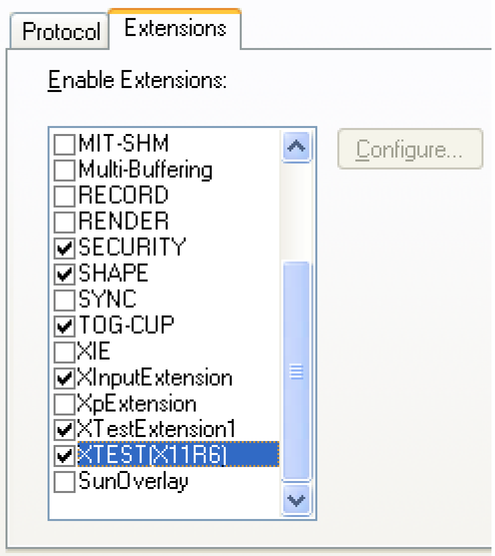
```
XTEST extension not installed on this X server: Error 0
```

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:



Enable Extensions:

☐ MIT-SHM
☐ Multi-Buffering
☐ RECORD
☐ RENDER
☑ SECURITY
☑ SHAPE
☐ SYNC
☑ TOG-CUP
☐ XIE
☑ XInputExtension
☐ XpExtension
☑ XTestExtension1
☑ XTEST(X11R6)
☐ SunOverlay

5. Restart the X Server and run the installer again.

Could not execl 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 execl 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=$UP module_type=form module=FORM_OR_MENU
```

RIB Errors

At random times, the RIB will get certain errors such as GETNXT(?,?,?,?,?,?,?) 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.

H

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: jdbc:oracle:oci:@<sid>

<sid>: system identifier for the database

Example: jdbc:oracle:oci:@mysid

Thin Client Syntax: jdbc:oracle:thin:@<host>:<port>:<sid>

<host>: hostname of the database server

<port>: database listener port

<sid>: system identifier for the database

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

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>/<manged_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.

I

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 1 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 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:

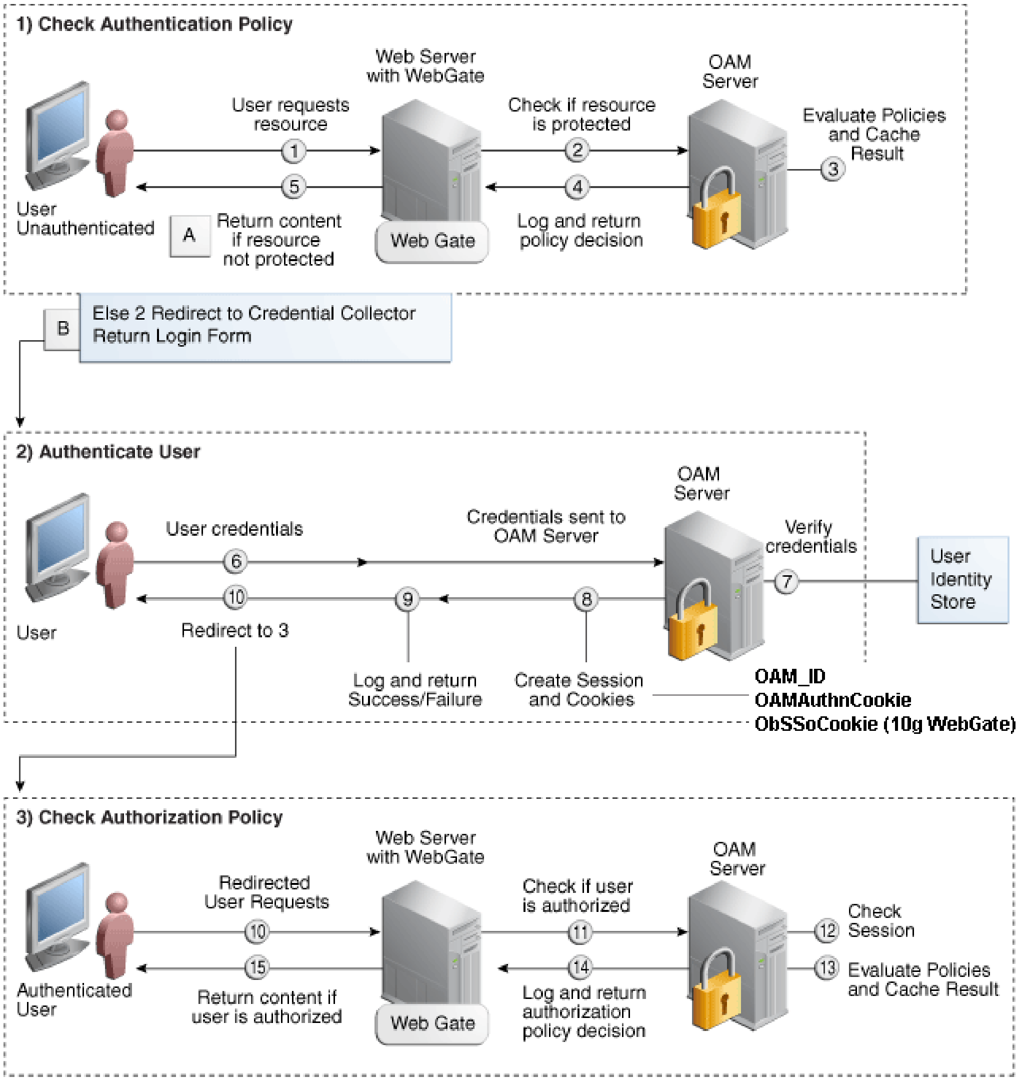
- The Oracle Access Manager (OAM) server, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle Access Manager Agent associated with the Web application, which verifies 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


1) Check Authentication Policy
Web Server with WebGate
OAM Server
User requests resource
Check if resource is protected
Evaluate Policies and Cache Result
1
2
3
4
5
User Unauthenticated
A
Return content if resource not protected
Web Gate
Log and return policy decision
B
Else 2 Redirect to Credential Collector Return Login Form
2) Authenticate User
OAM Server
User credentials
Credentials sent to OAM Server
Verify credentials
6
7
8
9
10
User Identity Store
User
Redirect to 3
Log and return Success/Failure
Create Session and Cookies
OAM_ID
OAMAuthnCookie
ObSSoCookie (10g WebGate)
3) Check Authorization Policy
Web Server with WebGate
OAM Server
Redirected User Requests
Check if user is authorized
Check Session
10
11
12
13
14
15
Evaluate Policies and Cache Result
Authenticated User
Return content if user is authorized
Web Gate
Log and return authorization policy decision

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 11gR1 (11.1.1.6). The ODSM application can be used for user and realm management within OID.
2. Oracle Access Manager 11gR1 (11.1.1.5) has to be installed and configured.
3. Additional midtier instances (such as Oracle Forms 11g) for Oracle Retail applications based on Oracle Forms technologies (such as RMS). These instances must be registered with the OAM installed in step 2.
4. Additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities must be registered with OAM installed in step 2.

Infrastructure Installation and Configuration

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

OID User Data

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

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

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

User Management

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

ODSM

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

LDIF Scripts

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

User Data Synchronization

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

J

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 were 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 /@dvols29_wms01user

Steps to Set up a pl/sql Wallet

1. Create a new directory called wallet under your folder structure where you run RWMS batch.
   a. cd /projects/rwms14.0/dev/forms
   b. mkdir wallet
   c. chmod 755 wallet

   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 WALLET_LOCATION is on line 1 in the file

```
WALLET_LOCATION =   (SOURCE =     (METHOD = FILE)     (METHOD_DATA =
(DIRECTORY =  /projects/rwms14.0/dev/wallet)) )
SQLNET.WALLET_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 two custom entries that are only for use with the wallet (ex: sqlplus /@dvols29_wms01user).

```
ifile = /u00/oracle/product/11.2.0.3/network/admin/tnsnames.ora

dvols29_wms01user =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))

dvols29_wms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))
```

> **Note:** It is important 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
   /projects/rwms14.0/dev/forms/wallet
   ```

   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 dvols29_wms01user wms01user oracle

6. Test connectivity. The ORACLE_HOME used with the wallet must be the same version or higher then what the wallet was created with.

   ```
   $ export TNS_ADMIN=/projects/rwms14.0/dev/forms

   $ sqlplus /@dvols29_wms01user

   SQL*Plus: Release 11.2.0.3.0 - Production on Fri Jan 14 12:53:39 2011

   Copyright (c) 1982, 2009, Oracle.  All Rights Reserved.


   Connected to:
   Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
   With the Partitioning, Real Application Clusters, Automatic Storage Management
   and Real Application Testing options

   SQL> show user
   USER is "wms01user"
   ```

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:

  ```
  dvols29_wms01user
  ```

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

  Returns value of the entry:

  ```
  wms01user
  ```

  ```
  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=/projects/rwms14.0/dev/forms
export ORACLE_WALLET_ENTRY=dvols29_wms01user
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 /projects/rwms14/extras).
   - mkdir javawallet
2. Ensure that retail-public-security-api.jar is unjarred and that the JAR is specified in the path.
3. Locate the save_credential.sh script. [Usually present in retail-public-security-api/bin directory].
4. The application wallet requires two entries for BI Publisher reports to work. The first one is user name/password@ConnectString for RWMS Database. The second one is username/password for BI Publisher. See examples in the next step.

5. Create a wallet using the below command:
   - ./save_credential.sh -u <USERNAME> -a <ALIASNAME> -p <PARTITION NAME> -l <WALLET PATH>

     Example: ./save_credential.sh -u wms01user -a RWMS14_USER -p rwms14 -l /projects/rwms14/extras/javawallet

     It will ask for password [Twice] and the wallet will be created.

     For e.x. password will be retek@sid

     ./save_credential.sh -u Administator-a BI_ALIAS -p rwms14 -l /projects/rwms14/extras/javawallet

     Here password could be Administrator.
6. 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
7. The entries in .env file will look like as shown in example below.

   rwms14inst.env

```
--------------
TNS_ADMIN=/projects/rwms14/dev/forms/wallet
RWMS_WALLET_PATH=/projects/rwms14/extras/javawallet
RWMS_WALLET_PARTITION=rwms14
RWMS_WALLET_LOGON=TRUE
RWMS_BI_PWD=BI_ALIAS
RWMS_DB_CONNECT= RWMS14_USER

CLASSPATH=$CLASSPATH:/projects/rwms14/extras/wmsSecurity.jar
```

K

Appendix: Setting Up Password Stores with wallets/credential stores

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

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

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

About Database Password Stores and Oracle Wallet

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

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

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

```
sqlplus /@db_username
```

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

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

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

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

Setting Up Password Stores for Database User Accounts

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

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

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

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

1. Create a wallet using the following command:

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

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

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

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

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

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

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

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

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

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

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

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

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

Setting up Wallets for Database User Accounts

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. Create the wallet files. These are empty initially.

   a. Ensure you are in the intended location.

      ```
      $ pwd
      /projects/rms14/dev/.wallet
      ```

   b. Create the wallet files.

      ```
      $ mkstore -wrl . –create
      ```

   c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.

   d. Enter the password again.

   Two wallet files are created from the above command:

   – ewallet.p12

   – cwallet.sso

5. Create the wallet entry that associates the user name and password to the custom tns alias that was setup in the wallet's tnsnames.ora file.

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

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

6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.

```
$ export TNS_ADMIN=/projects/rms14/dev/.wallet /* This is very import to use wallet to
point at the alternate tnsnames.ora created in this example */

$ sqlplus /@dvols29_rms01user

SQL*Plus: Release 11

Connected to:
Oracle Database 11g

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

Running batch programs or shell scripts would be similar:

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

Set the UP unix variable to help with some compiles :

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

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

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

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

- Change the password for a credential on wallet

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

- List the wallet credential entries

  ```
  mkstore –wrl . –list
  ```

  This command returns values such as the following.

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

- View the details of a wallet entry

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

  Returns the value of the entry:

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

  Returns 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 `$MMHOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env.`
   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     - `export RETL_WALLET_ALIAS="retl_java_rms01user"`
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     - `export ORACLE_WALLET_ALIAS="dvols29_rms01user"`
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     - `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"`

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

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

For Java applications, consider the following:

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

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

- Two main scripts are related to this script in the folder for more generic wallet operations: dump_credentials.sh and save_credential.sh.
- If you have not installed the application yet, you can unzip the application zip file and view these scripts in <app>/application/retail-public-security-api/bin.
- Example:
- /u00/webadmin/reim14/application/retail-public-security-api/bin

update-<ALIAS>.sh

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

Usage:

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

Example:

```
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/retail-
public-security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass in.
mspdev71:[1034WLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/RPMDomain/retail/rpm14/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/product/10.3.x/WLS/user_projects/domains/REIMDomain/retail/reim14/
config

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

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

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

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

save_credential.sh

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

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

Example:

```
mspdv351:[1036_WLS] /u00/webadmin/mock14_testing/rtil/rtil/application/retail-public-
security-api/bin> save_credential.sh -l wallet_test -a myalias -p mypartition -u myuser

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

Enter password:
Verify password:
```

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

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

Usage

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

How does the Wallet Relate to the Application?

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

Reim.properties code sample:

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

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

How does the Wallet Relate to Java Batch Program use?

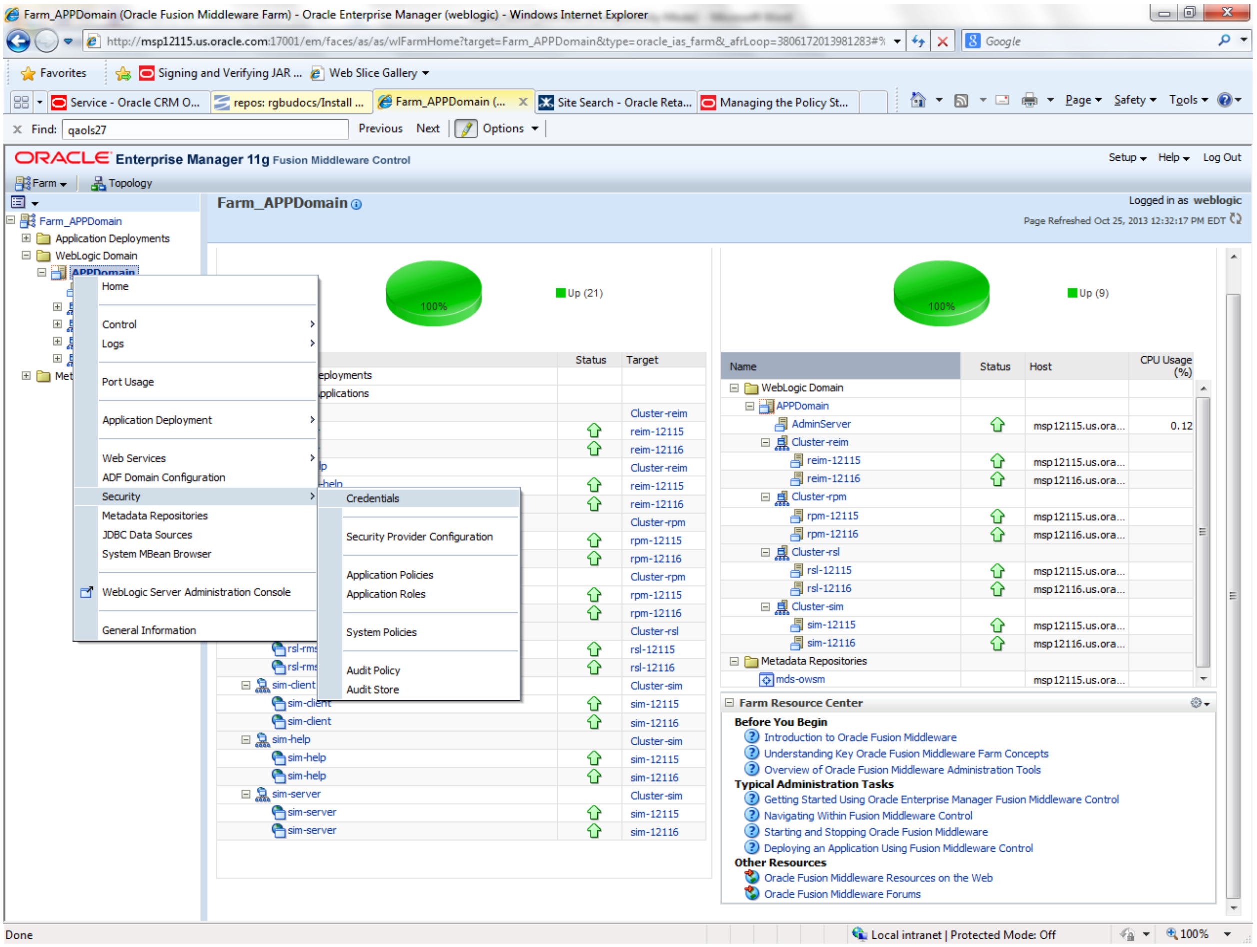
Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser RMS01APP, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments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

Database Credential Store Administration

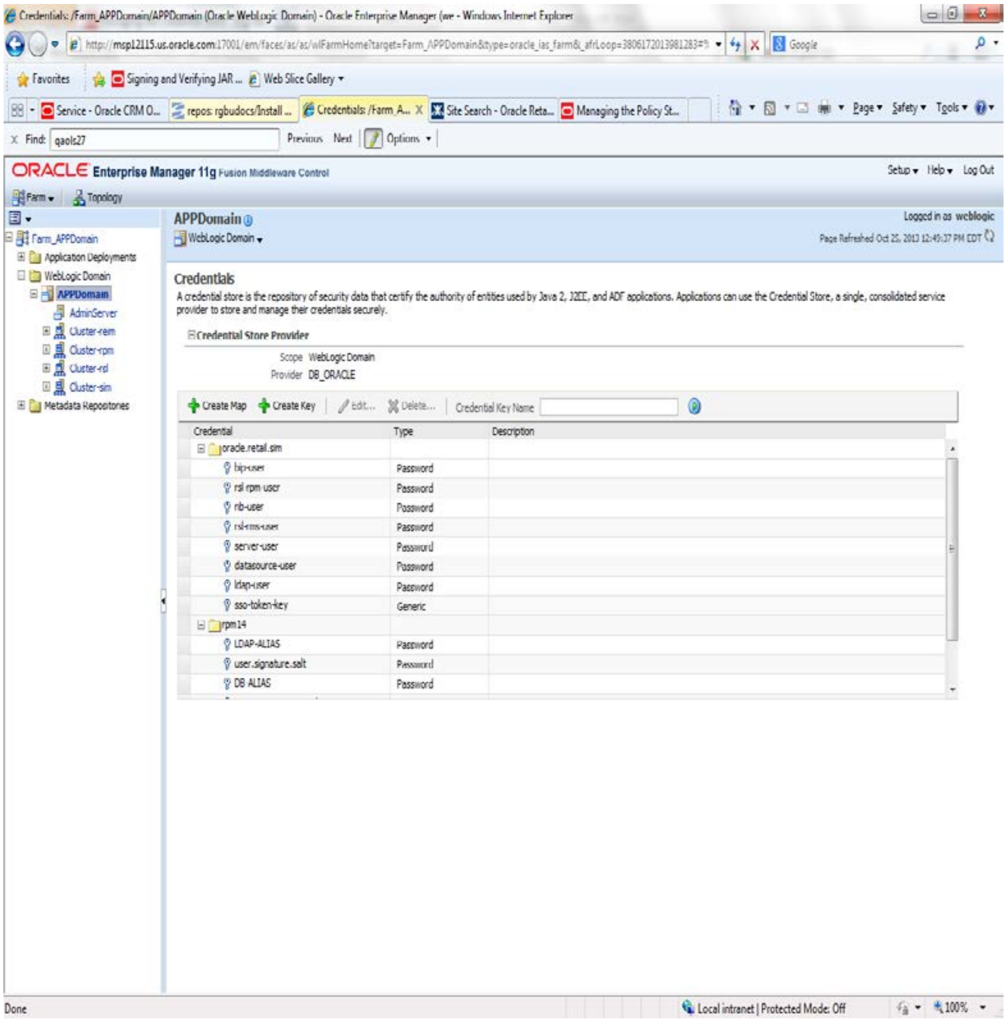
ORACLE Retail 14.0 brings something new into the password stores. A domain level database credential store. This is used in RPM login processing, SIM login processing, and Allocation login processing and policy information for application permission. Setting up the database credential store is addressed in the RPM, SIM, and Alloc 14.0 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.


Credentials: /Farm_APPDomain/APPDomain (Oracle WebLogic Domain) - Oracle Enterprise Manager (we - Windows Internet Explorer
http://msp12115.us.oracle.com:17001/em/faces/as/as/wlFarmHome?target=Farm_APPDomain&type=oracle_ias_farm&_afrLoop=3806172013981283#%
Google
Favorites
Signing and Verifying JAR ...
Web Slice Gallery
Service - Oracle CRM O...
repos: rgbudocs/Install ...
Credentials: /Farm_A...
Site Search - Oracle Reta...
Managing the Policy St...
Page
Safety
Tools
Find: qaols27
Previous
Next
Options
ORACLE Enterprise Manager 11g Fusion Middleware Control
Setup
Help
Log Out
Farm
Topology
Farm_APPDomain
Application Deployments
WebLogic Domain
APPDomain
AdminServer
Cluster-rem
Cluster-rpm
Cluster-rsl
Cluster-sim
Metadata Repositories
APPDomain
WebLogic Domain
Logged in as weblogic
Page Refreshed Oct 25, 2013 12:49:37 PM EDT
Create Map
Create Key
Edit...
Delete...
Credential Key Name
Done
Local intranet | Protected Mode: Off
100%

Credentials

A credential store is the repository of security data that certify the authority of entities used by Java 2, J2EE, and ADF applications. Applications can use the Credential Store, a single, consolidated service provider to store and manage their credentials securely.

Credential Store Provider

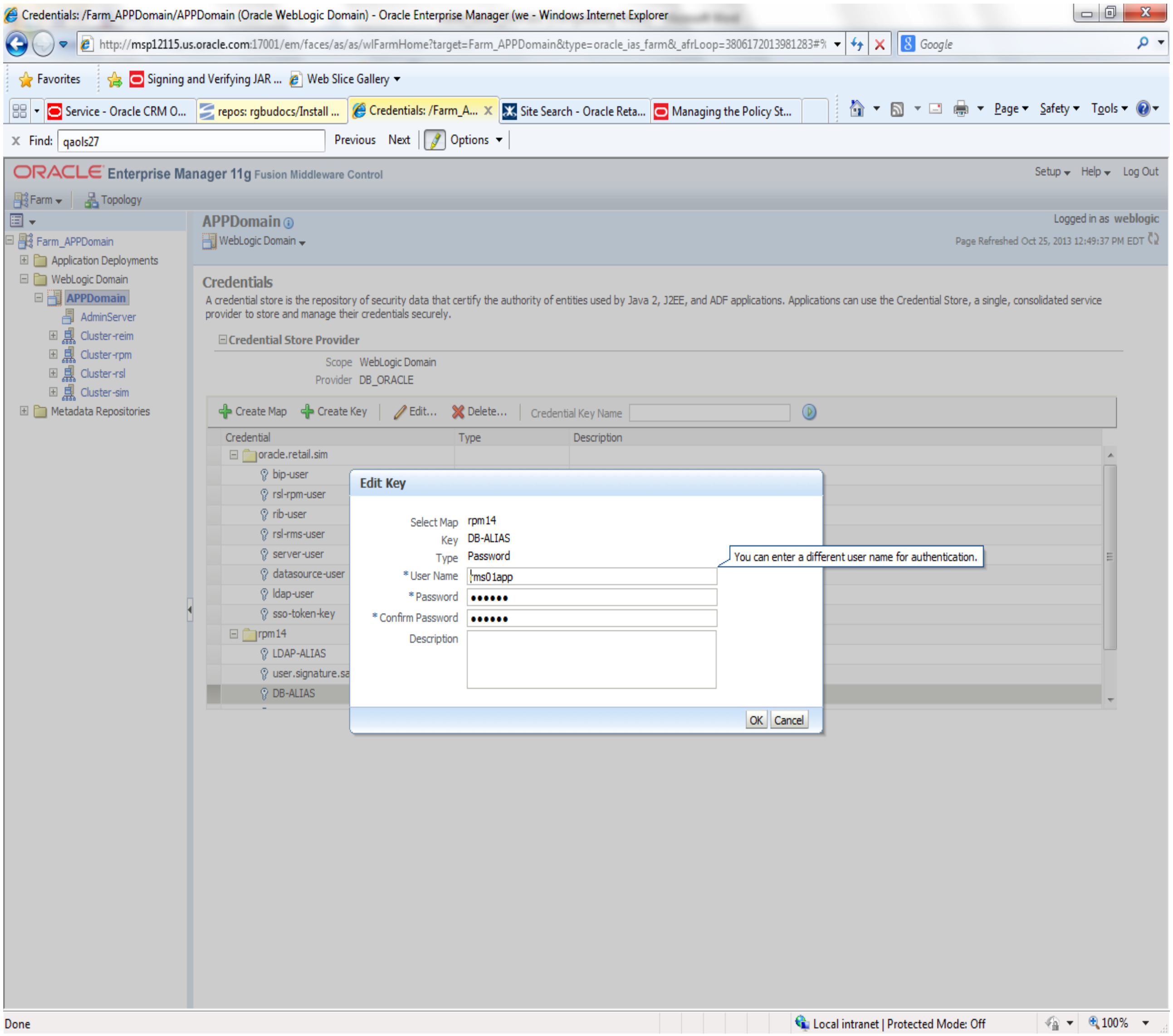
Scope WebLogic Domain

Provider DB_ORACLE

Credential	Type	Description
oracle.retail.sim		
bip-user	Password	
rsl rpm user	Password	
rib-user	Password	
rsl-rms-user	Password	
server-user	Password	
datasource-user	Password	
ldap-user	Password	
sso-token-key	Generic	
rpm14		
LDAP-ALIAS	Password	
user.signature.salt	Password	
DB ALIAS	Password	

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

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


Credentials: /Farm_APPDomain/APPDomain (Oracle WebLogic Domain) - Oracle Enterprise Manager (we - Windows Internet Explorer
http://msp12115.us.oracle.com:17001/em/faces/as/as/wlFarmHome?target=Farm_APPDomain&type=oracle_ias_farm&_afrLoop=3806172013981283#%
Google
Favorites
Signing and Verifying JAR ...
Web Slice Gallery
Service - Oracle CRM O...
repos: rgbudocs/Install ...
Credentials: /Farm_A...
Site Search - Oracle Reta...
Managing the Policy St...
Page
Safety
Tools
Find: qaols27
Previous
Next
Options
ORACLE Enterprise Manager 11g Fusion Middleware Control
Setup
Help
Log Out
Farm
Topology
Farm_APPDomain
Application Deployments
WebLogic Domain
APPDomain
AdminServer
Cluster-reim
Cluster-rpm
Cluster-rsl
Cluster-sim
Metadata Repositories
APPDomain
WebLogic Domain
Logged in as weblogic
Page Refreshed Oct 25, 2013 12:49:37 PM EDT
Credentials
A credential store is the repository of security data that certify the authority of entities used by Java 2, J2EE, and ADF applications. Applications can use the Credential Store, a single, consolidated service provider to store and manage their credentials securely.
Credential Store Provider
Create Map
Create Key
Edit...
Delete...
Credential Key Name
Credential
Type
Description
oracle.retail.sim
bip-user
rsl-rpm-user
rib-user
rsl-rms-user
server-user
datasource-user
ldap-user
sso-token-key
rpm14
LDAP-ALIAS
user.signature.sa
DB-ALIAS
Edit Key
You can enter a different user name for authentication.
OK
Cancel
Done
Local intranet | Protected Mode: Off
100%

Scope WebLogic Domain
Provider DB_ORACLE

Select Map rpm14
Key DB-ALIAS
Type Password
* User Name rms01app
* Password ••••••
* Confirm Password ••••••
Description

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

Managing Credentials with WSLT/OPSS Scripts

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

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

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

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

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

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

```
msp12115:[1036_APP] /u00/webadmin/product/wls_apps/oracle_common/bin> ./orapki
wallet display –wallet
/u00/webadmin/product/wls_apps/user_projects/domains/APPDomain/config/fmwconfig
```

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

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

```
Oracle PKI Tool : Version 11.1.1.6.0
Copyright (c) 2004, 2011, Oracle and/or its affiliates. All rights reserved.

Requested Certificates:
User Certificates:
Oracle Secret Store entries:
rpm14@#3#@DB-ALIAS
rpm14@#3#@LDAP-ALIAS
rpm14@#3#@RETAIL.USER
rpm14@#3#@user.signature.salt
rpm14@#3#@user.signature.secretkey
rpm14@#3#@WEBLOGIC-ALIAS
rpm14@#3#@WLS-ALIAS
Trusted Certificates:
```

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

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

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

listCred

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

Script Mode Syntax

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

Interactive Mode Syntax

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

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

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

Examples of Use:

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

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

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

```
msp12115:[1036_APP] /u00/webadmin/product/wls_apps/oracle_common/common/bin>
sh wlst.sh

Initializing WebLogic Scripting Tool (WLST)...

Welcome to WebLogic Server Administration Scripting Shell


wls:/offline> connect('weblogic','password123','msp12115.us.oracle.com:17001')
Connecting to t3://msp12115.us.oracle.com:17001 with userid weblogic ...
Successfully connected to Admin Server 'AdminServer' that belongs to domain 'APPDomain'.

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

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

updateCred

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

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

createCred

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

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

deleteCred

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

Script Mode Syntax

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

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

modifyBootStrapCredential

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

This script is available in interactive mode only.

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

Any output regarding the audit service can be disregarded.

addBootStrapCredential

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

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

This script is available in interactive mode only.

Interactive Mode Syntax

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

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

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

Example of Use:

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

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

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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (MMHOME)>/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile, execution	Installer	n/a	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile	Installer	n/a	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	n/a	<Db_Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rwms schema owner>	Compile forms, execute batch	Installer	n/a	Alias hard-coded by installer
RPM app	DB	<RPM batch install dir>/.wallet	n/a	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	RPM plsql and sqlldr batches
RWMS auto-login	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	<RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms14inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	n/a	Alias hard-coded by installer

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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm14	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			rpm14	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	
			rpm14	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm14	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm14	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name, ex: reim14>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name, ex: reim14>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name, ex: reim14>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name, ex: reim14>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbat-alias	
Alloc app	DB credential store		Map=alloc 14 or what you called the app at install time	Many for login and policies					<weblogic domain home>/config/fmwconfig/jps-config.xml has info on the credential store. This directory also has the domain cwallet.sso file. The bootstrap directory under this directory has bootstrap cwallet.sso file.
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/config							Each alias must be unique
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms schema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
			<installed app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc batch user alias>	<SYSTEM_ADMINISTRATOR>	Batch use	Installer	alloc14	
RSL app	JAVA	<RSL INSTALL DIR>/rsl-rms/security/config							Each alias must be unique
			rsl-rsm	<rsl weblogic user alias>	<rsl weblogic user name>	App use	Installer	weblogic-alias	
			rsl-rsm	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
SIM app	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>/config	<installed sim app name>	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			<installed sim app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			<installed sim app name>	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application

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

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

L

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 MMHOME environment variable
- $MMHOME/orpatch/bin/orcompile –a RWMS –t FORMS

Usage:

orcompile -a <app> -t <type>

Potential Apps and Types:

RWMS   => DB,DB-ADF,FORMS

M

Appendix: Web Browser Configuration

This appendix provides information on configuring Internet Mozilla Firefox Web browsers for operation with RWMS.

Mozilla Firefox Version 17.0

Mozilla Firefox supports a JRE sandboxed environment in a different manner to Internet Explorer with additional plug-ins to support specific versions of the Java Console for the JREs. Firefox includes the accurate JPI versions. The formsweb.cfg file defines how Web browsers handle Forms applets. The browser is instructed to use the Java Platform Interface (JPI) for applets using the following entry in the formsweb.cfg file:

```
jpi_mimetype=application/x-java-applet;jpi-version=1.6.0_12
```

Update this entry to the following to make Firefox work:

```
jpi_mimetype=application/x-java-applet
```

Note: No server restart is required for this to work.

This does not impact the operation of Internet Explorer Web browsers configured as described in the previous section.

Firefox sets the plug-in interfaces it uses, so by removing the version details it can now use the relevant JRE installed. You must also configure the Firefox Web browser to ignore warnings about pages that use low grade security in the Options menu under the Security tab (in a similar way to Internet Explorer).

N

Appendix: Installation Order

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

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

Enterprise Installation Order

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

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

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

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

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

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

13. Oracle Retail Predictive Application Server (RPAS)
14. Oracle Retail Demand Forecasting (RDF)
15. Oracle Retail Category Management (CM)
16. Oracle Retail Modeling Engine (ORME)
17. Oracle Retail Assortment Space Optimization (OASO)

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