Oracle® Retail Warehouse Management System

Installation Guide
Release 13.2.5
E37579-02

February 2013

ORACLE

Oracle® Retail Warehouse Management System Installation Guide, Release 13.2.5

Primary Author: Donna Linde

Contributors: Nathan Young

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

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

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

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

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

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

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

Contents

Send Us Your Comments

Oracle Retail Warehouse Management System Installation Guide, Release 13.2.5

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 13.2.5 documentation set:

- *Oracle Retail Warehouse Management System Implementation Guide*
- *Oracle Retail Warehouse Management System Radio Frequency User Guide*
- *Oracle Retail Warehouse Management System Release Notes*
- *Oracle Retail Warehouse Management System User Interface User Guide*

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.2) or a later patch release (for example, 13.2.5). If you are installing the base release and additional patch and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

Conventions

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

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

1

Preinstallation Tasks

This chapter includes steps to complete before installation.

Check for the Current Version of the Installation Guide

Corrected versions of Oracle Retail installation guides may be published whenever critical corrections are required. For critical corrections, the rerelease of an installation guide may not be attached to a release; the document will simply be replaced on the Oracle Technology Network Web site.

Before you begin installation, check to be sure that you have the most recent version of this installation guide. Oracle Retail installation guides are available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technology/documentation/oracle_retail.html

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

If a more recent version of this installation guide is available, that version supersedes all previous versions. Only use the newest version for your installation.

Check 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 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 6.1, 7.1 (Actual hardware or LPARs) ▪ Solaris 10, 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) **Other components:** ▪ Perl compiler 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 Release 1 (11.1.1.6). Options are: ▪ Oracle Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 5 for x86-64 (Actual hardware or Oracle virtual machine). ▪ AIX 6.1, 7.1 (Actual hardware or LPARs) ▪ Solaris 10, 11 SPARC (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware, HPVM, or vPars)
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.6) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.6) ▪ Oracle Forms Services 11g Release 2 (11.1.2.0) ▪ Java: JDK 1.6.0+ 64 bit or ▪ Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only. **IMPORTANT:** If there is an existing WebLogic installation on the server, you must upgrade to WebLogic 10.3.6. All middleware components associated with WebLogic server 10.3.6 should be upgraded to 11.1.1.6 and ORACLE Forms to be upgraded to 11gR2 forms (11.1.2.0). Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed. **Optional (SSO required)** ▪ Oracle Internet Directory 10gR3 (10.1.4) optionally with Oracle Single Sign-On 10gR3 (10.1.4) or ▪ Oracle Identity Management 11gR1 (11.1.1.6) under Weblogic 10.3.6 optionally with Oracle Single Sign-On 10gR3 (10.1.4) or ▪ Oracle Identity Management 11gR1 (11.1.1.6) optionally with Oracle Access Manager 11gR1 (11.1.1.5). Must have separate WebLogic 10.3.5 for Oracle Access Manager 11g. **Other components:** ▪ Oracle BI Publisher 10g (10.1.3.4)

Verify Single Sign-On

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

If Single Sign-On is to be used, verify the Oracle Internet Directory (OID) 10gR3 version 10.1.4 or Oracle Identity Management (OIM/IDM) 11gR1 version 11.1.1.6 has been installed along with the components listed in the above Application Server requirements section . Verify the HTTP Server is registered with the Oracle Access Manager (OAM) 11gR1 as a partner application.

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
Display resolution	1024x768 or higher
Processor	2.6GHz or higher
Memory	1GByte or higher
Networking	intranet with at least 10Mbps data rate
Oracle (Sun) Java Runtime Environment	1.6.0_22+
Browser	Microsoft Internet Explorer version 8.0 or 9.0, Mozilla Firefox 3.6.2.3 or 10.0

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:
>
> - Symbol MC9090 - (Hand Held)
> - Motorola VC5090 - (Truck Mount)
> - Motorola WT4090 – (Wrist mount)

Supported Oracle Retail Products

Requirement	Version
Oracle Retail Merchandising System (RMS)	13.2.5
Oracle Retail Store Inventory Management (SIM)	13.2.5

Supported Oracle Retail Integration Technologies

Integration Technology	Version
Oracle Retail Integration Bus (RIB)	13.2.5

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. It is suggested that if you do use OCI connections, the Oracle Retail products database be configured in the tnsnames.ora file used by the Oracle Application Server installations.

Clustering for WebLogic Server 10.3.6 is managed as an Active-Active cluster accessed through a Load Balancer. It is suggested that a Web Tier 11.1.1.6 installation be configured to reflect all application server installations. Validation has been completed utilizing a RAC 11.2.0.3 Oracle Internet Directory database with the WebLogic 10.3.6 cluster.

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 (11.1) Part Number B28254-07

3

Patch Installation

The database portion of RWMS can be upgraded from any 13.2.x release. This guide details the steps needed to perform a patch installation of RWMS. For additional information on the upgrade, see the My Oracle Support document, *Oracle Retail Upgrade Guide* (ID 1073414.1).

The Oracle Retail Upgrade Guide describes the approach that this Oracle Retail application takes for the upgrading process, as well as this product's upgrade assumptions and considerations.

The following chapters document the patch process:

- Chapter 4, Patch RWMS Database Installation
- Chapter 5, Patch Application Server Installation Tasks

4

Patch RWMS Database Installation

This section describes how to update the RWMS database schema to release 13.2.5. It is assumed that your existing RWMS database schema is on at least release 13.2.

Note: Shut down any applications that may be using the RWMS schema (for example RIB) before applying a schema patch.

Note: If you are installing multiple patches, you must run the database schema patch installer separately for each patch. For example, if upgrading from 13.2 to 13.2.5, the installer should first be run to install the 13.2.3 scripts, followed by 13.2.3.1, 13.2.4, and 13.2.5.

Patch RWMS Database using the Patch Installer

The RWMS 13.2.5 database schema patch installer should be used to apply the RWMS patch.

Before you apply the RWMS 13.2.5 patch:

- Make a backup of all your objects and database schema.
- Determine which patches and hot fix bundles have already been installed.
- Review the enclosed RWMS 13.2.5 Release Notes (rwms-1325-rn.pdf).
- Review each of the enclosed defect documents.
- Make sure any applications that connect to the RWMS schema are shut down. This includes RIB and anything else that could be using the schema.

Create Staging Directory for RWMS Database Schema Files

To create a staging directory for RWMS database schema files, complete the following steps.

1. Log in to the database server as a user that can connect to the RWMS database.
2. Create a staging directory for the RWMS 13.2.5 Patch. There should be a minimum of 140 MB disk space available in this location.
3. Copy the rwms1325dbpatch.zip file from the RWMS 13.2.5 release to the staging directory. This is referred to as DB_PATCH_DIR when patching a database schema.
4. Change directories to DB_PATCH_DIR and extract the rwms1325dbpatch.zip file. This creates a RWMS/dbschemapatch subdirectory under DB_PATCH_DIR.

Optional Security Step from Previous Patch

If you ran the optional revoke_rwms_admin_privs.sql script to secure the permissions for the RWMS schema that was included in the 13.2.4 patch, you must grant additional permissions before running the database installation.

To grant necessary install permissions to the RWMS schema, run the following script as the database sys user.

```
DB_PATCH_DIR/rwms/dbschemapatch/util/grant_rwms_admin_privs.sql
```

For more information, see "Appendix: Optional Security."

Run the RWMS Database Schema Patch Installer

Note: See "Appendix: RWMS Database Patch Installer Screens" for details about screens and fields in the RWMS database schema patch installer.

1. Change directories to DB_PATCH_DIR/rwms/dbschemapatch.
2. Source the oraenv script to set up the Oracle environment variables (such as ORACLE_HOME, ORACLE_SID, PATH).

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
NLS_LANG	Locale setting for Oracle database client	NLS_LANG=AMERICAN_AMERICA.UTF8 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 details.
5. If the patch installer has already been run in this location you may wish to back up the ant.install.properties file. The settings from the RWMS 13.2.5 patch installation will be refreshed with the latest input every time the installer runs.

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

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

   install.sh [text | silent]

7. On the Apply a Patch page, provide the path to the corresponding patch you are applying. This path will be DB_PATCH_DIR/rwms/dbschemapatch/rwms-dbpatch/<version>/.
8. After the installer is complete, you can check its log file: rwms-install-dbschema.<timestamp>.log.
9. 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`

10. If you are applying multiple patches (for example, if you are going from 13.2 to 13.2.5), rerun the installer. On the Apply a Patch page, select the next patch in the sequence. Repeat this step until you have installed all required patches for the desired patch level.

    Note: For more information, see "Appendix: Optional Security."

5

Patch Application Server Installation Tasks

There are two different methods to use for installing the RWMS 13.2.5 Application. Option 1 uses the installer to apply the patch. Option 2 compiles the RWMS forms directly.

This section describes how to upgrade the RMWS application from 13.2.0 to 13.2.5.

Be sure to review the *Oracle Retail Warehouse Management Release Notes* for the 13.2.5 release.

> **IMPORTANT**: If there is an existing WebLogic installation on the server, you must upgrade WebLogic to WebLogic 10.3.6. All middleware components associated with WebLogic server should be upgraded to 11.1.1.6.
>
> Back up the weblogic.policy file ($WLS_HOME/wlserver_10.3/server/lib) before upgrading your WebLogic server, because this file could be overwritten. Copy over the weblogic.policy backup file after the WebLogic upgrade is finished and the post patching installation steps are completed.

Option 1: Use Application Installer to Patch

If you already have RWMS 13.2 installed, do not use the RWMS 13.2 application installer to apply this patch because customizations will be overwritten. Instead, compile the forms manually. Go to the section "Option 2: Compile RWMS Forms Manually."

If you are choosing to install RWMS 13.2 and all subsequent patches up to 13.2.5 at the same time and you have not run the RWMS 13.2 application installer, the RWMS 13.2 application installer can be used to install and compile all the forms in 13.2 and all subsequent patches up to 13.2.5 at one time.

Create Staging Directory for RWMS Application Files

To create a staging directory for RWMS application patch files, complete the following steps.

1. Log on to your application server as a user with read and write access to the WebLogic files.
2. If you do not already have one, create a staging directory for the RWMS application installation software (13.2). There should be a minimum of 600 MB disk space available in this location.
3. Copy the file rwms13application.zip from the RWMS 13.2 release to staging directory. This will be referred to as STAGING_DIR when installing application software and reports.
4. Change directories to STAGING_DIR and extract the file rwms13application.zip. This will create an rwms/application subdirectory under STAGING_DIR.
5. Create a staging directory for the RWMS Application patch (which contains all patches from 13.2.x through 13.2.5). There should be a minimum of 150 MB disk space available in this location.

6. Copy the file rwms1325apppatch.zip from the RWMS 13.2.5 release to staging directory. This will be referred to as APP_PATCH_DIR when installing application software and reports.
7. Change directories to APP_PATCH_DIR and extract the file rwms1325apppatch.zip. This creates an app-patch subdirectory under APP_PATCH_DIR.

Custom Modules

Custom source can be provided by the user in a folder named APP_PATCH_DIR/app-patch/patch-util/custom. The source code in this folder is applied last, after all patches have been applied.

Run the Installer Patching Utility

Note: This utility automates the copying of forms and libraries from the APP_PATCH_DIR to the STAGING_DIR.

To run the installer patching utility, complete the following steps.

1. Set the JAVA_HOME environment variable to point to a JDK.
2. Set the ANT_HOME environment variable to point to an Ant installation. There is one included with the RWMS installer that can be used for this.

   ANT_HOME=<INSTALL_DIR>/rwms/application/ant

   export ANT_HOME
3. Change directories to APP_PATCH_DIR/app-patch/patch-util/
4. Modify the patch.properties file. Set the installer.dir and patch.to.version properties.

Variable	Description
installer.dir	The directory where the installer files are located under STAGING_DIR. Example: /opt/rwms/application. This directory will contain the install.sh file.
patch.to.version	The version to which you want to patch. Example: 13.2.5

5. Run the patch.sh script. This script will copy the files from each patch up to the patch specified in the **patch.to.version** property and generate a full installer package containing all forms up to the current patch..

Run the RWMS Application Installer

To run the RWMS application installer, do the following.

1. Log on to your application server as a user with read and write access to the WebLogic files.
2. Change directories to STAGING_DIR/rwms/application. This directory was created when the rwms13application.zip file was expanded under STAGING_DIR,
3. Set and export the following environment variables.

Variable	Description	Example
DOMAIN_HOME	The location where Forms 11.1.2 domain has been installed.	DOMAIN_HOME= /u00/webadmin/product/10.3.x/WLS_Forms/user_projects/domains/ClassicDomain/ export DOMAIN_HOME
WLS_INSTANCE	The name of the managed server that contains Oracle Forms. The name will appear under $DOMAIN_HOME/config/fmwconfig/servers	WLS_INSTANCE=WLS_FORMS export WLS_INSTANCE
ORACLE_SID	The database/SID where the RWMS schema resides.	ORACLE_SID=mydb export ORACLE_SID
NLS_LANG	Locale setting for Oracle database client.	NLS_LANG=AMERICAN_AMERICA.UTF8 export NLS_LANG
JAVA_HOME	Location of Java. Usually the same Java which is being used by Weblogic. Java choices are: JDK 1.6.0+ 64 bit Or Jrockit 1.6 R28 build or later, within the 1.6 code line. 64 bit. For Linux and Solaris OS only.	JAVA_HOME= /u00/webadmin/java/jdk1.6.0_18 export JAVA_HOME or JAVA_HOME=/u00/webadmin/java/jrockit1.6R28 export JAVA_HOME
DISPLAY	Address and port of X server on desktop system of user running install. Required for forms application installer.	DISPLAY=<IP address>:0 export DISPLAY

4. To install the RWMS application you must use an X server such as Exceed and have set the DISPLAY environment variable. The installer does not continue otherwise.
5. Run the install.sh script to start the installer.

> **Note:** Below are the usage details for install.sh. The typical usage for GUI mode is no arguments.
>
> ./install.sh [text | silent]

6. The Installer automatically sets additional environment variables based on the values of the environment variables set in Step 3. At the end of the preinstallation checks a summary will print, containing these new environment variables:

   Example:

   MW_HOME=/u00/webadmin/product/10.3.x/WLS_Forms

   ORACLE_HOME=/u00/webadmin/product/10.3.x/WLS_Forms/as_1

   ORACLE_INSTANCE=/u00/webadmin/product/10.3.x/WLS_Forms/asinst_1

   DOMAIN_HOME=/u00/webadmin/product/10.3.x/WLS_Forms/user_projects/domains/ClassicDomain

   WLS_INSTANCE=WLS_FORMS

   ORACLE_SID=mydb

   JAVA_HOME=/u00/webadmin/java/jdk1.6.0_18 (This should be set to the JDK being used by the WebLogic server.)

   Verify that these environment variables are correct. If any of them are incorrect, verify that the WebLogic shell scripts that set them are configured properly. Check the following scripts:

   ```
   $DOMAIN_HOME/bin/setDomainEnv.sh
   $WEBLOGIC_HOME/wlserver_10.3/common/bin/commEnv.sh
   ```

   Example:

   /u00/webadmin/product/10.3.x/WLS_Forms/user_projects/domains/ClassicDomain/bin/setDomainEnv.sh

   /u00/webadmin/product/10.3.x/WLS_Forms/wlserver_10.3/common/bin/commEnv.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.

Note: You may see the following warning repeated during installation:

```
     [exec] Warning! One or more of your selected
locales are not available.
     [exec] Please invoke the commands "locale" and
"locale -a" to verify your
     [exec] selections and the available locales.
     [exec]
     [exec] Continuing processing using the "C" locale.
```

Or

```
     [exec] couldn't set locale correctly
```

This warning can be ignored.

7. After the installation is complete, check the log file to verify a successful installation: INSTALL_DIR/base/log/rwms.app.install.<timestamp>.log. The installer may not fail if forms have failed to compile, so verify by scanning the file for forms compilation errors (for example, "Compilation errors have occurred"). If none are found, the installation was successful.
8. 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`

   Note: The object counts performed by the installer may be off due to the patch adding or removing modules. The installer warnings about this can be ignored
9. 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_Forms/user_projects/domains/ClassicDomain/config/
fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/forms/registry/ora
cle/forms/registry/Registry.dat
/u00/webadmin/product/10.3.x/WLS_Forms/user_projects/domains/ClassicDomain/config/
fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/formsweb.cfg

Have the WebLogic administrator stop WLS_FORMS and ohs1,
copy everything in /home/oretail/rwms132/install/post
to /u00/webadmin/product/10.3.x/WLS_Forms to update the files
and then start WLS_FORMS and ohs1
for the changes to take effect.

example: cp -R * /u00/webadmin/product/10.3.x/WLS_Forms
```

Resolving Errors Encountered During Application Installation

In the event a form or menu does not compile, go to INSTALL_DIR/base/error to identify the objects that did compile. To manually recompile the object, run INSTALL_DIR/base/forms.profile and run the following command:

```
# frmcmp.sh userid=$UP module_type=form module=FORM_OR_MENU
```

You can also safely rerun the installer to see if the form compiles.

> **Note:** If you rerun the installer, and choose to check the **Configure WebLogic** box in the installer screens, you may need to clean up duplicate entries in the WebLogic formsweb.cfg file.

RWMS Reports Copied by the Application Installer

The application installer copies RWMS report files to INSTALL_DIR/base/reports. These files should be installed into BI Publisher as documented in the chapter, RWMS Reports Installation—Patch.

Option 2: Compile RWMS Forms Manually

To compile the RWMS forms manually requires the following steps.

1. Log on to the application server as the user that has write access to WebLogic.
2. Set the DISPLAY variable to the IP address plus ":0.0" (for example, 10.1.1.1:0.0) of the application server.

> **Note:** ORACLE_HOME is the location where Oracle Forms 11gR2 has been installed
> ORACLE_INSTANCE is the location where WebLogic has been installed and contains the executables to compile forms.

All OS Platforms

Set the following variables:

- PATH=$ORACLE_HOME/bin:$ORACLE_HOME/opmn/bin:$ORACLE_HOME/dcm/bin:INSTALL_DIR/base/forms_scripts:$PATH
- CLASSPATH=$ORACLE_HOME/jlib/importer:$ORACLE_HOME/jlib/debugger.jar:$ORACLE_HOME/jlib/utj.jar:$ORACLE_HOME/jlib/ewt3.jar:$ORACLE_HOME/jlib/share.jar:$ORACLE_HOME/jlib/dfc.jar:$ORACLE_HOME/jlib/help4.jar:$ORACLE_HOME/jlib/oracle_ice.jar:$ORACLE_HOME/jlib/jewt4.jar
- FORMS_BUILDER_CLASSPATH=$CLASSPATH
- FORMS_PATH=INSTALL_DIR/base/toolset/bin:INSTALL_DIR/base/forms/bin:$ORACLE_HOME/forms
- REPORTS_PATH=INSTALL_DIR/base/reports/bin:$ORACLE_HOME/forms
- TK_UNKNOWN=$ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
- UP=<RWMS schema owner>/<RWMS schema password>@<RWMS database>

  Note: See "Appendix: Setting Up Password Stores with Oracle Wallet" for how to set up database wallet.

  Note: Verify that TNS is set up correctly by using the UP variable to successfully log in to the RWMS 13 schema.

  Example: /u00/oracle> sqlplus $UP

- AIX
  - LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/jdk/jre/lib
  - LIBPATH=$LD_LIBRARY_PATH
- Linux
  - LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/jdk/jre/lib

Install RWMS Forms

Complete the following steps to install the forms and scripts.

1. Make a backup copy of the INSTALL_DIR/base/forms directories.

   For the version number of the patch you are installing, navigate to the corresponding directory in APP_PATCH_DIR and copy the contents of the following directories into INSTALL_DIR. Depending on the patch, these directories may not exist.

   - Copy all forms (fmb) in APP_PATCH_DIR/ app-patch/<version>/forms/src to the INSTALL_DIR/base/forms/src directory.
   - Copy all scripts in APP_PATCH_DIR/ app-patch/<version>/forms/bin to the INSTALL_DIR/base/forms/bin directory.
2. Change directories to INSTALL_DIR/base/forms/src.
3. Run forms.fmb.sh (in INSTALL_DIR/base/forms/src) to generate RWMS runtime forms – .fmx's.
   - Move all .fmx's to the INSTALL_DIR/base/forms/bin directory

Delete Obsolete Files

The following forms are obsolete as of the 13.2.5 release and should be deleted from INSTALL_DIR.

```
INSTALL_DIR/base/forms/src/hh_container_checking_s.fmb
INSTALL_DIR/base/forms/bin/hh_container_checking_s.fmx
INSTALL_DIR/base/forms/src/wip_code_seq_editor_s.fmb
INSTALL_DIR/base/forms/bin/wip_code_seq_editor_s.fmx
```

Configure WebLogic 10.3.6 for RWMS

> **Note:** The following steps are required to configure WebLogic for RWMS. You may not need these steps if you are upgrading from a previous installation.

To configure WebLogic 10.3.6 for RWMS, complete the following steps.

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

1. Make a copy of the file **default.env** file located in ORACLE_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config and rename it to a relevant file name (for example, rwms.env).
2. Modify the new rwms.env file by appending the location of the RWMS forms modules to the FORMS_PATH variable setting, and by adding the NLS_DATE_FORMAT and NLS_LANG variables to the end of this file. Additionally, the variable FORMS_REJECT_GO_DISABLED_ITEM=FALSE and FORMS_USERNAME_CASESENSITIVE=1 must also be added to the rwms.env file due to the changes between older version of Oracle Forms and WebLogic 10.3.6. For example,

   ```
   FORMS_PATH=/u00/rwms/forms/bin:/u00/oracle/product/10.3.x_WLS/WLS/
              asinst_1/FormsComponent/forms
   NLS_DATE_FORMAT=DD-MON-RR
   NLS_LANG=AMERICAN_AMERICA.UTF8
   FORMS_USERNAME_CASESENSITIVE=1
   FORMS_REJECT_GO_DISABLED_ITEM=FALSE
   ```

   Add other variables `ORACLE_RWMS_EXTRAS`, `RWMS_FORMS_SERVER`, and `RF_LAUNCH_VALUE_ALIAS` at the end of the **rwms.env** file.

   For example,

   ```
   ORACLE_RWMS_EXTRAS=/u00/rwms/extras
   RWMS_FORMS_SERVER=http://<WebLogic_forms_server:portnumber>/
   RF_LAUNCH_VALUE_ALIAS=/forms/rwms/rf_launch/
   ```

3. Complete the following steps to set up the RF launch module:
   a. Copy the **rf_launch** folder from the APP_DIR/rwms to the following location:
      `ORACLE_HOME/user_projects/domains/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_11.1.2/e18uoi/war/rwms/`
   b. Update the **rwms_rf_menu.htm** file with the correct URLs for the variables `hh_device`, `tm_device`, `wr_device`, and `exit_script`.
      For example,

```
var hh_device = "http://server:9001/forms/frmservlet?config=rwms132wls_hh";
var tm_device = “http://server:9001/forms/frmservlet?config=rwms132wls_tm";
var wr_device = "http://server:9001/forms/frmservlet?config=rwms132wls_wr";
var exit_script = "http://server:9001/forms/rwms/rf_launch/close.htm”;
```

4. Start the opmn processes at $ORACLE_HOME/asinst_1/bin using the following command:
   `opmnctl startall`
5. Make an entry in the file ORACLE_HOME/as_1/network/admin/tnsnames.ora for the Oracle 11g database that was created in Chapter2 (where the RWMS 13 database schema was installed). Appendix D contains a sample tnsnames.ora file entry for an Oracle 11g database. Refer to the sample tnsnames.ora file entry for an Oracle database in Appendix D or the example available in the file ORACLE_HOME/network/admin/tnsnames.ora.
6. Modify the file formsweb.cfg located at ORACLE_HOME/user_projects/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/ to include an RWMS environment section at the end of this file. In the following example, square brackets highlight a separate environment in this file. In the RWMS environment section of the formsweb.cfg file, set up the variables `envfile` (from step 2 above), `width`, `height`, and `separateFrame` applet parameters, and the starting form for the RMS application. For example,

```
[rwms]
        envfile=rwms.env
        width=950
        height=685
        separateFrame=true
        lookAndFeel=Oracle
        colorScheme=swan
        archive=frmall.jar,rwms-icons.jar
        imagebase=codebase
        form=logon_scr.fmx
```

   If Oracle Single Sign-On is to be used with RWMS, do the following.
   - Set ssoMode to true.
   - If Resource Access Descriptors are allowed to be dynamically created, set ssoDynamicResourceCreate to true.
     For example,

```
[rwms]
        envfile=rwms.env
        width=950
        height=685
        separateFrame=true
        lookAndFeel=Oracle
        colorScheme=swan
        archive=frmall.jar,rwms-icons.jar
        imagebase=codebase
```

```
form=logon_scr.fmx
ssoMode=true
```

To allow the **rwms-icons.jar** file to be picked up in your forms, the following line also needs to be added to the formsweb.cfg file:

```
# Forms applet archive setting for other clients (Sun Java Plugin,
Appletviewer, etc)
archive=frmall.jar, rwms-icons.jar
```

7. If NLS_LANG is NOT set in the ORACLE_HOME/forms/server/rwms.env, copy the RWMS keyboard-mapping file from INSTALL_DIR/base/sample_files/fmrweb.res to ORACLE_HOME/forms/admin/resource/US.

   If NLS_LANG is set in the ORACLE_HOME/forms/server/rwms.env file, copy the RWMS keyboard-mapping file from INSTALL_DIR/base/sample_files/fmrweb_utf8.res to ORACLE_HOME/forms/admin/resource/US.

8. Load RWMS by entering the following URL in a Web browser.

   Note: Prior to testing, the JRE 1.6.0_22+ plug-in needs to be installed on the client machine. The plug-in can be downloaded from the Java Web site (http://www.oracle.com/technetwork/java/index.html).

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

   Where,

   - server is the name or IP address of the server where Oracle Forms Services 11g Release 2 (11.1.2) is running.
   - port is the "Listen" value in ORACLE_HOME/user_projects/domains/ClassicDomain/config/config.xml It is the port listed for WLS_FORMS.
   - env is the name of the environment in brackets in formsweb.cfg (from step 5 above). For example, `http://server:9001/forms/frmservlet?config=rwmswls`.

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

The first time RWMS is accessed, the user is prompted with the following security warning. Click **Yes**.

Security Warning Message

If Single Sign-On is not used, or if a Resource Access Descriptor has not been set up for RMS for this user and ssoDynamicResourceCreate is set to true, then the RMS logon form appears. Enter the relevant user name, password, and connect string information in the corresponding fields:

- Username – The RWMS Schema owner or additional Oracle user created.
- Password – Associated password for the user name.
- Connect String – Oracle database created in Chapter 1.
  For example,

  Username: RWMS13DEV

  Password: retek

  Connect String: prod_db1

9. On the RWMS logon form, enter the relevant user name, password, and connect string in the corresponding fields:
   - Username – RWMS Schema owner or additional Oracle user created.
   - Password – Password associated with the user name.
   - Connect String – Oracle database created in Chapter 1.
     For example,

     Username: RWMS13DEV

     Password: retek

     Connect String: prod_db1

Clustered Installations – Post-Installation Steps

If you are installing the RWMS application to a clustered Oracle Application Server environment, there are some extra steps you need to take to complete the installation. In these instructions, the application server node with the ORACLE_HOME you used for the RWMS application installer is referred to as master node. All other nodes are referred to as remote nodes.

To complete the RWMS forms application install, the installer provided new versions of formsweb.cfg and the newly-created env files for the new RWMS installation. The env files should be copied from the master node to the remote node(s). The entries added to formsweb.cfg for these new environments should be copied from the master node to the remote nodes.

> **Note:** Do not copy the entire formsweb.cfg file from one node to another. Only copy the RWMS entries appended to this file by the installer. There is node-specific information in this file that is different between ORACLE_HOME installations.

Install the Online Help

To install the online help, do the following.

1. Log on to the WebLogic instance where the online help will be installed.
2. Select the WLS_FORMS server, and click **Next**.
3. Click **Deployments**.
4. Click the **Install** button.
5. In the **Path** field, select the rwms-help.ear file**.** Leave the **Install this deployment as an application** check box selected and click **Next**.
6. Select the **rwms-help-server** created in step 2, and click **Next**.
7. Select deploy application
8. Select proper target
9. Select default settings and Activate changes (upper left)

Install rwms-icons.jar

If the patch contains a new rwms-icons.jar, follow these steps to install it:

1. Copy the rwms-icons.jar from APP_PATCH_DIR/app-patch/<version>/web_html/ and overwrite the one in $ORACLE_HOME/forms/java/
2. Bounce the WebLogic managed server that contains Forms (for example:, WLS_FORMS).

Remove rwms13unix.conf from the forms configuration

If it exists, remove rwms13unix.conf from the forms configuration ($DOMAIN_HOME/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/rwms132inst/rwms13unix.conf). To remove this file:

1. Delete rwms13unix.conf

```
rm /u00/webadmin/product/10.3.x/WLS_Forms/user_projects/domains/ClassicDomain/
config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.2/config/rwms132inst
/rwms13unix.conf
```

2. Take a backup of runtime httpd.conf.

```
cp /u00/webadmin/product/10.3.x/WLS_Forms/asinst_1/config/OHS/ohs1/httpd.conf
/u00/webadmin/product/10.3.x/WLS_Forms/asinst_1/config/OHS/ohs1/httpd.conf.
```

3. vi http.conf
4. Delete the line with reference to rms13unix.conf
5. Save the file.
6. Restart the OHS.

Test the RWMS Application

Oracle Retail provides test cases that allow you to smoke test your installation. See the My Oracle Support document, *Oracle Retail Merchandising Installation Test Cases* (ID 1277131.1.1).

6

RWMS Reports Installation—Patch

RWMS Reports are included in the RWMS Application patch: rwms1325apppatch.zip in the reports directories.

> **Note:** If the BI Publisher application 10g is already deployed to a bipublisher managed server in WebLogic, you can directly go to the section, "Manually Copy Reports to Install Directory." If not, continue to the section, "BI Server Component Installation Tasks."

BI Server Component Installation Tasks

Oracle BI Publisher is used as the main RMS and RWMS 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 10.3.4. One deployment of BI Publisher can be used for both the RMS and RWMS reports.

If you are installing BI Publisher as a part the Oracle BI EE suite, refer to the appropriate Fusion Middleware guides for the installation of the product in a WebLogic server environment. Otherwise, you must perform the steps described in this section to deploy Oracle BI Publisher as a standalone Web application into a WebLogic server environment.

Installation Process Overview

To install BI Publisher server as a standalone Web application in a WebLogic server involves, you must do the following:

1. Locate the correct and generic version of xmlpserver.war from the BI Publisher source media.
2. Create an exploded directory from the xmlpserver.war file, where the BI server installation will reside on the WebLogic server.
3. Deploy BI Publisher into the WebLogic application server instance.
4. Configure the BI Publisher repository.
5. Optionally, install additional fonts into the JRE of the WebLogic server's JDK if you are planning to develop reports that are directly rendered by BI Publisher.

After BI Publisher is installed, you must do the following:

6. Set up the RMS BI Publisher Report Templates produced for RWMS.
7. Create the BI Publisher scheduler schema on the database server (required to send and schedule reports).
8. Set up for the RWMS application specific configuration files to integrate BI Publisher.

Extract the BI Server Web Archive from the Source Media

The BI Server components must be extracted from the source installation media. If you have downloaded the source distribution of Oracle BI EE, you must locate the BI Publisher source directory from the media.

The BI Publisher install media contains the following:

- BI Publisher server application
- BI Publisher runtime libraries
- BI Publisher fonts
- BI Publisher desktop tools
- Documentation: User's Guide, "Javadocs," demos, and samples

Individual components are located in the directory structure as follows:

Directory	Component
<BI_MEDIA>/fonts	Font files
<BI_MEDIA>/generic	xmlpserver.war (for non-oc4j application servers) xmlpserver.ear (for non-oc4j application servers)
<BI_MEDIA>/oc4j	xmlpserver.ear (for oc4j)
<BI_MEDIA>/XMLP	XMLP repository
<BI_MEDIA>/XMLP/DemoFiles	Data source for demo reports
<BI_MEDIA>/XMLP/Tools	Template Builder for Word Add-in

Complete the following steps.

1. Create the <BI_DEPLOYMENT> directory on the server and change directory into this directory.
   For example, assuming that /u00/webadmin is the root of the installation:
   ```
   mkdir /u00/webadmin/RWMS_BIP
   cd /u00/webadmin/RWMS_BIP
   ```
2. Locate the manual/generic/xmlpserver.war file from this directory structure and copy it to the <BI_DEPLOYMENT> directory on the server using a copy command with the following syntax:

   cp <BI_MEDIA>/manual/generic/xmlpserver.war <BI_DEPLOYMENT>

   For example,

   ```
   cp /tmp/BIPublisherSource/manual/xmlpserver.war /u00/webadmin/RWMS_BIP
   ```

Create an Exploded Directory for the Installation

You must create an exploded archive directory from this xmlpserver.war file. This will be the directory from which WebLogic will run the BI Server.

> **Note:** Do not deploy the xmlpserver.war or xmlpserver.ear file on the WebLogic Server by uploading it from the WebLogic console, because the console deploys the application (or Web module) in an archived file format.
>
> This is not recommended for BI Publisher configuration, because you must update WEB-INF/xmlp-server-config.xml manually before the deployment. To work around this issue, use an "exploded archive" directory.

Complete this task by following these steps:

1. Change directory to <BI_DEPLOYMENT> on the server.

   For example, assuming that /u00/webadmin is the root of the installation:

   ```
   cd /u00/webadmin/RWMS_BIP
   ```

2. By running the jar command with -x for extraction with the file xmlpserver.jar, create an exploded directory called "xmlpserver" within <BI_DEPLOYMENT>. This is the location where the application will be deployed in the WebLogic server.

   For example,

   ```
   mkdir xmlpserver
   cd xmlpserver
   jar -xvf /u00/webadmin/RWMS_BIP/xmlpserver/xmlpserver.war
   ```

   You should now have an exploded directory structure with <BI_DEPLOYMENT>/xmlpserver/ for the deployment.

   > **Note:** Any changes to the BI Publisher configuration files, (such as an update to the catalog path in the xmlp-server-config.xml file) must be done before deployment. For more information on catalogs, refer to BI Publisher documentation.

3. Delete the war file.

   For example,

   ```
   rm /u00/webadmin/RWMS_BIP/xmlpserver/xmlpserver.war
   ```

Configure the BI Publisher Repository and Installing Fonts

Before deploying BI Publisher in your WebLogic server, you must set up the BI Publisher repository and install the font files into the JVM used by the server.

To set up the repository, copy the <BI_MEDIA>/XMLP directory to <BI_REPOSITORY>.

For example,

```
cp -R /tmp/BIPublisherSource/XMLP /u00/webadmin/RWMS_BIP/xmlpserver
```

1. Assign appropriate permissions for the WebLogic server instance user with read, write, and execute permissions to enable the deployment of this directory structure to serve as a Web application.
2. Open the xmlp-server-config.xml file located in BI_HOME_DIR/xmlpserver/WEB-INF directory with a text editor.

   For example,

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <xmlpConfig xmlns="http://xmlns.oracle.com/oxp/xmlp">
       <resource>
           <file path="${oracle.home}/xdo/repository"/>
       </resource>
   </xmlpConfig>
   ```
3. Replace ${oracle.home}/xdo/repository with file path to the location where you copied the XMLP repository directory on your server.
   For example,

   ```
   <?xml version="1.0" encoding="UTF-8"?>
       <xmlpConfig xmlns="http://xmlns.oracle.com/oxp/xmlp">
       <resource>
           <file path="/u00/webadmin/RWMS_BIP/xmlpserver/XMLP"/>
       </resource>
   </xmlpConfig>
   ```
4. Save the xmlp-server-config.xml after updating it.
5. (Optional) Copy the font files from the <BI_MEDIA>/fonts in the installation media to the fonts directory of the Java Runtime Environment used by the WebLogic server for the deployment (represented by WLS_JAVA_HOME in the example below). This is an optional task for users who plan to deploy their own customized reports for RMS using BI Publisher directly.
   For example,

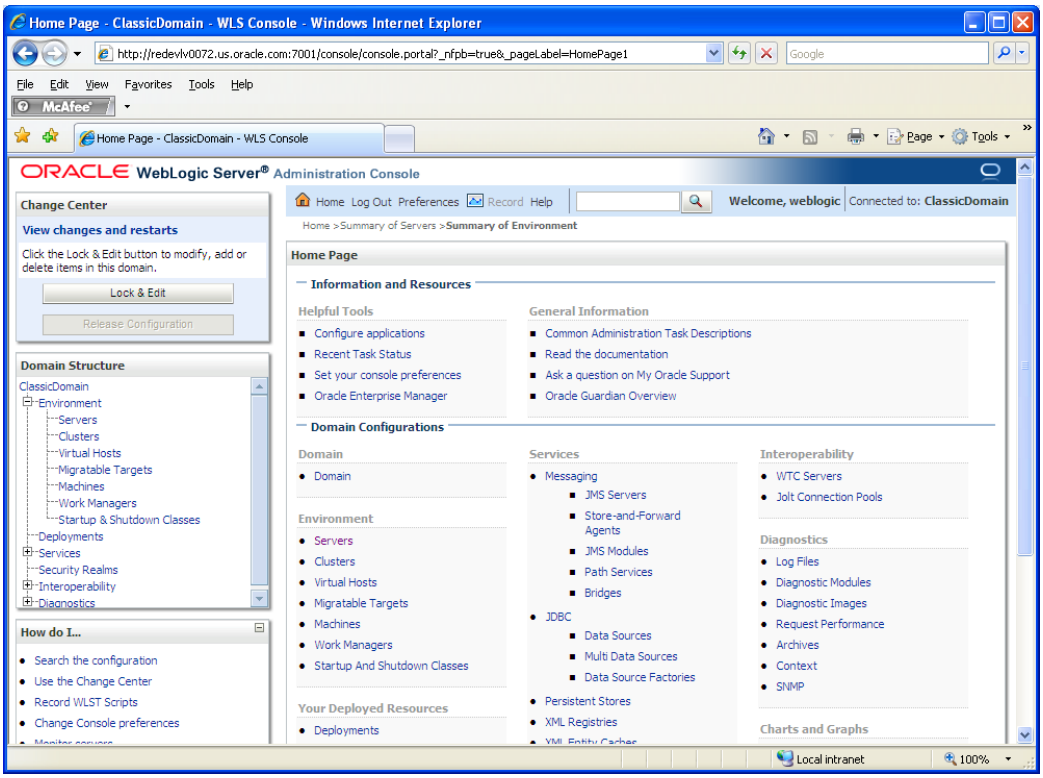
   ```
   cp - R /tmp/BIPublisherSource/fonts WLS_JAVA_HOME/jre/lib/fonts
   ```

   Note: This task requires that you restart the WLS server.

Install Managed Server in WebLogic

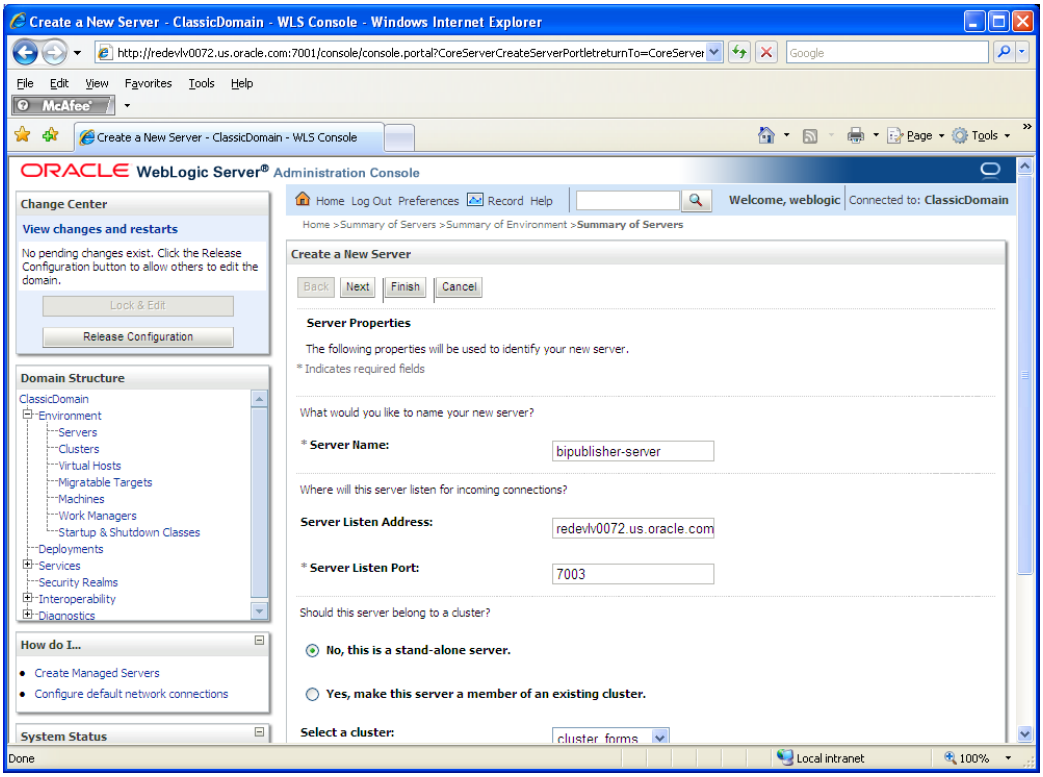
Before running the deployment of BI Application, you must install a managed server for deploying the BI application in WebLogic, if it was not created during the domain installation. Follow the steps below to install bipublisher managed server.

1. Log in to the administration console.



2. Click **Lock & Edit**.

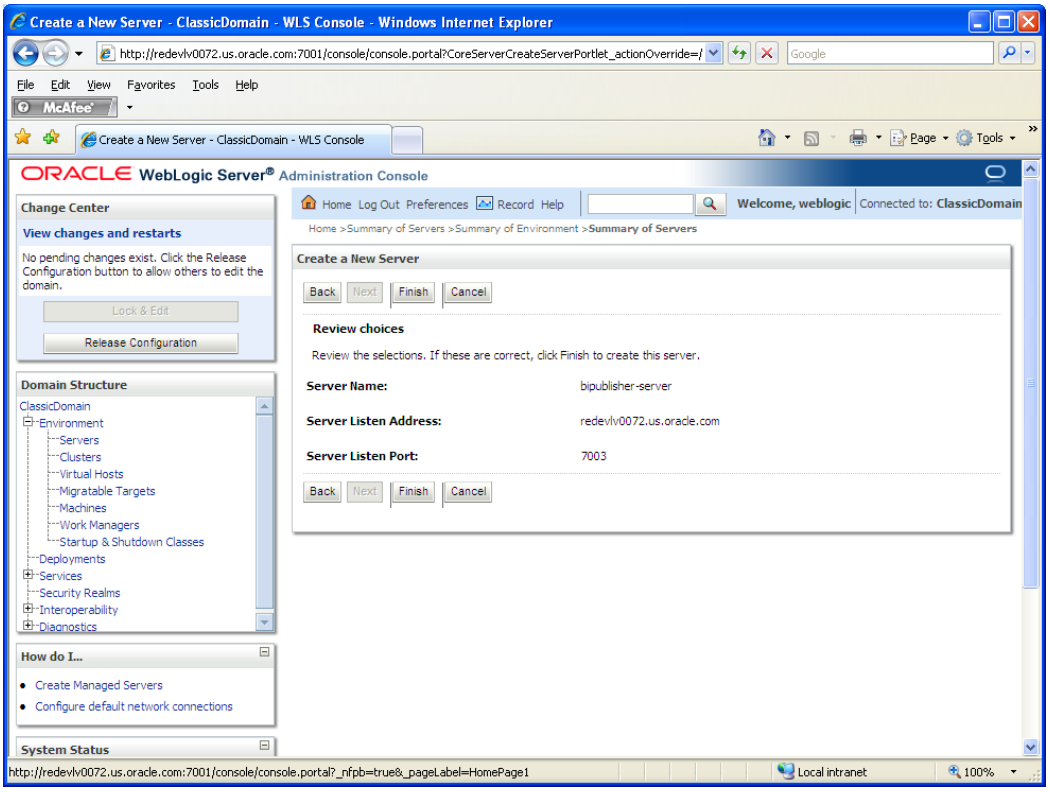
3. Navigate to Environment > Servers. Select the new tab for the servers on the right.



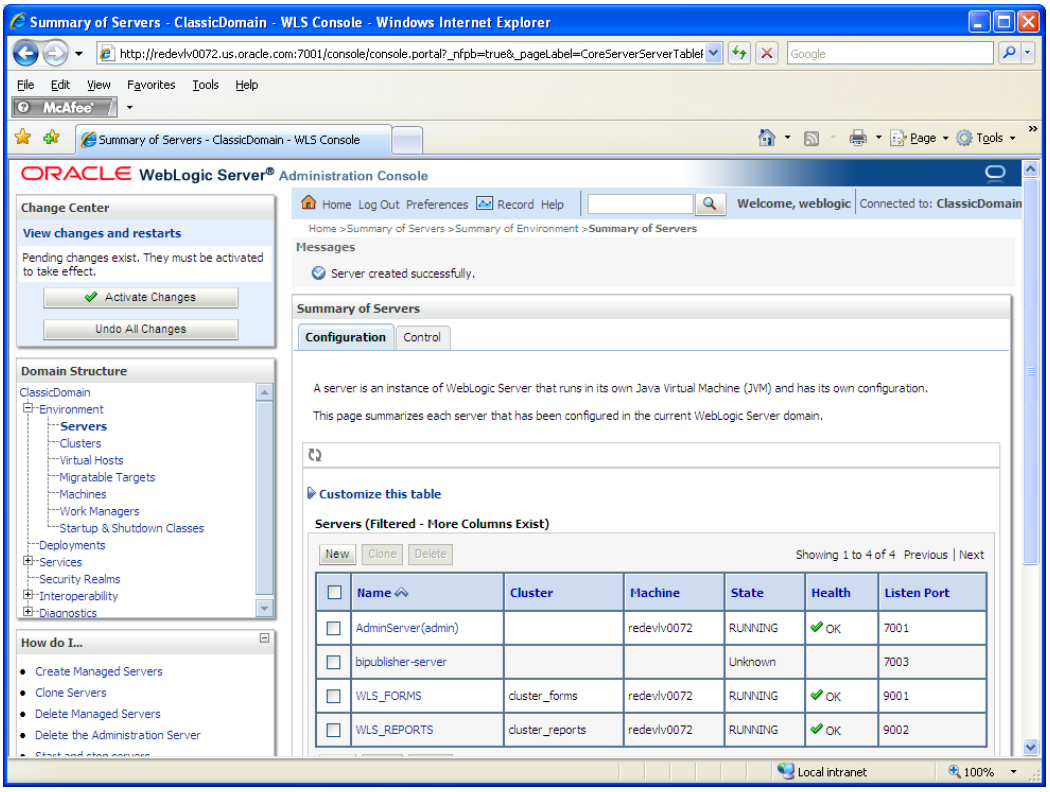
4. Set the following variables:
   - **Server Name**: This value should be specific to your targeted application (for example, bipublisher-server)
   - **Server Listen Address**: <weblogic server> (for example, redevlv0072.us.oracle.com)
   - **Server Listen Port**: A free port. Check for availability.

     A suggestion is to increment the AdminServer port by two and keep incrementing by two for each managed server (for example, 7003, 7005, 7007 and so on).

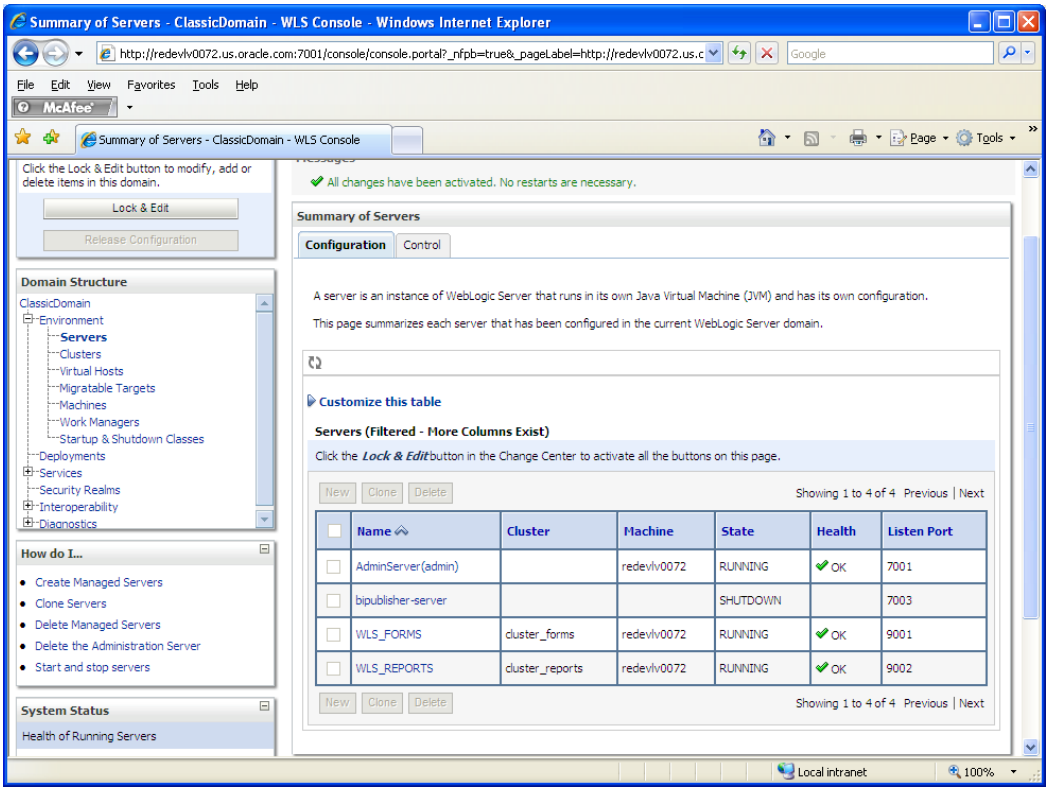
5. Click **Next**.


Create a New Server - ClassicDomain - WLS Console - Windows Internet Explorer
http://redevlv0072.us.oracle.com:7001/console/console.portal?CoreServerCreateServerPortlet_actionOverride=/
File Edit View Favorites Tools Help
Create a New Server - ClassicDomain - WLS Console
ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts
No pending changes exist. Click the Release Configuration button to allow others to edit the domain.
Lock & Edit
Release Configuration
Domain Structure
ClassicDomain
Environment
Servers
Clusters
Virtual Hosts
Migratable Targets
Machines
Work Managers
Startup & Shutdown Classes
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create Managed Servers
Configure default network connections
System Status
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: ClassicDomain
Home >Summary of Servers >Summary of Environment >Summary of Servers
Create a New Server
Back Next Finish Cancel
Review choices
Review the selections. If these are correct, click Finish to create this server.
Server Name: bipublisher-server
Server Listen Address: redevlv0072.us.oracle.com
Server Listen Port: 7003
Back Next Finish Cancel
http://redevlv0072.us.oracle.com:7001/console/console.portal?_nfpb=true&_pageLabel=HomePage1
Local intranet
100%

6. Click **Finish**.



7. Click **Activate Changes** on the left side. Once the changes are activated, the State of the bipublisher-server should change to SHUTDOWN status.



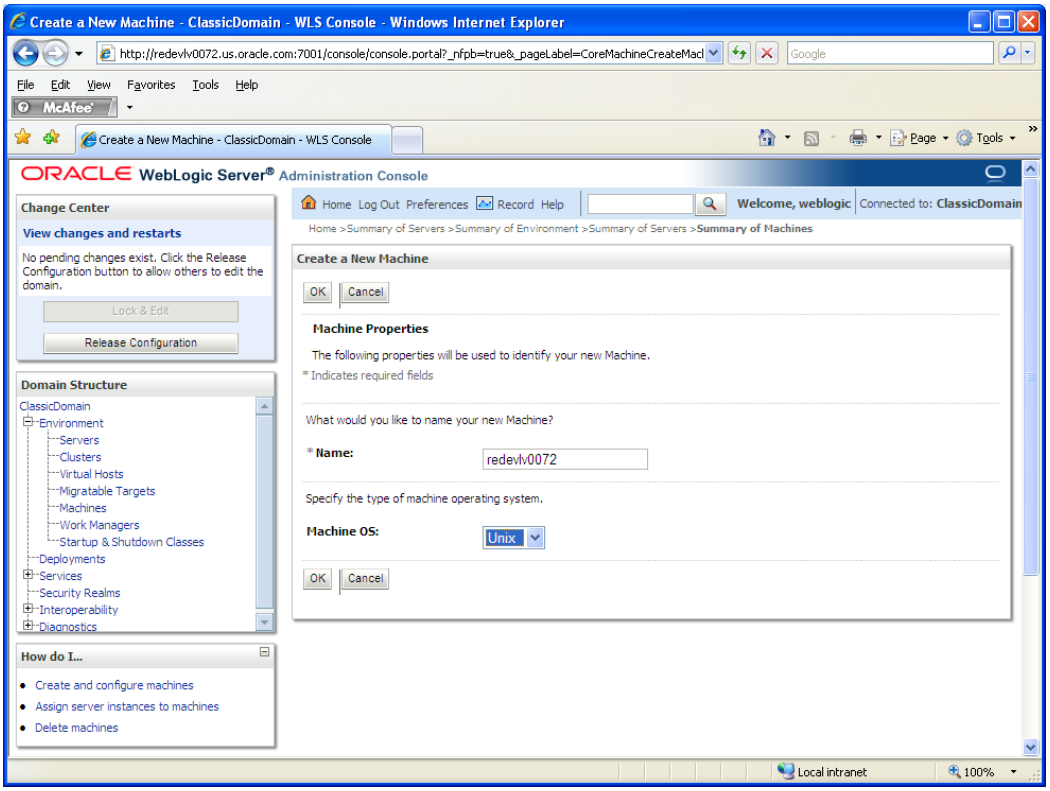
Install Node Manager

Install Node Manager if it was not created during domain install. Node Manager is required so that the managed servers can be started and stopped through the administration console. Only one node manager is needed per WebLogic installation.

1. Log in to the administration console.
2. Click **Lock & Edit**. Navigate to Environments->Machines. Click **New**.

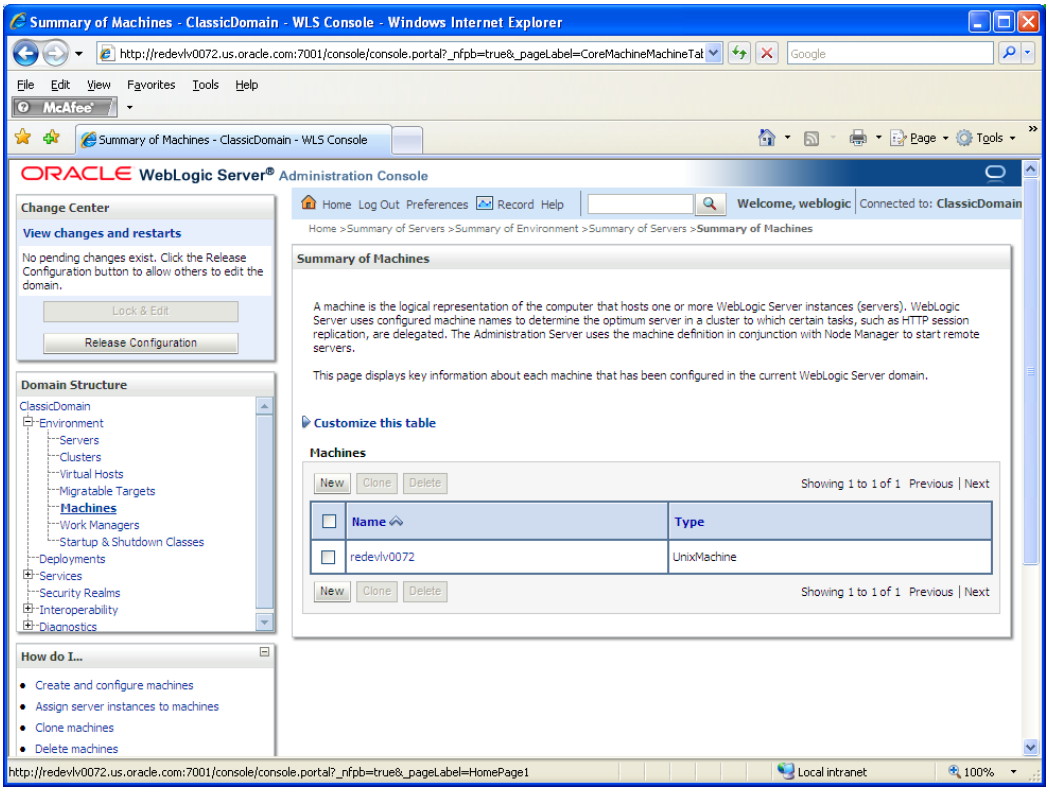
   The following page is displayed. Set the following variables:

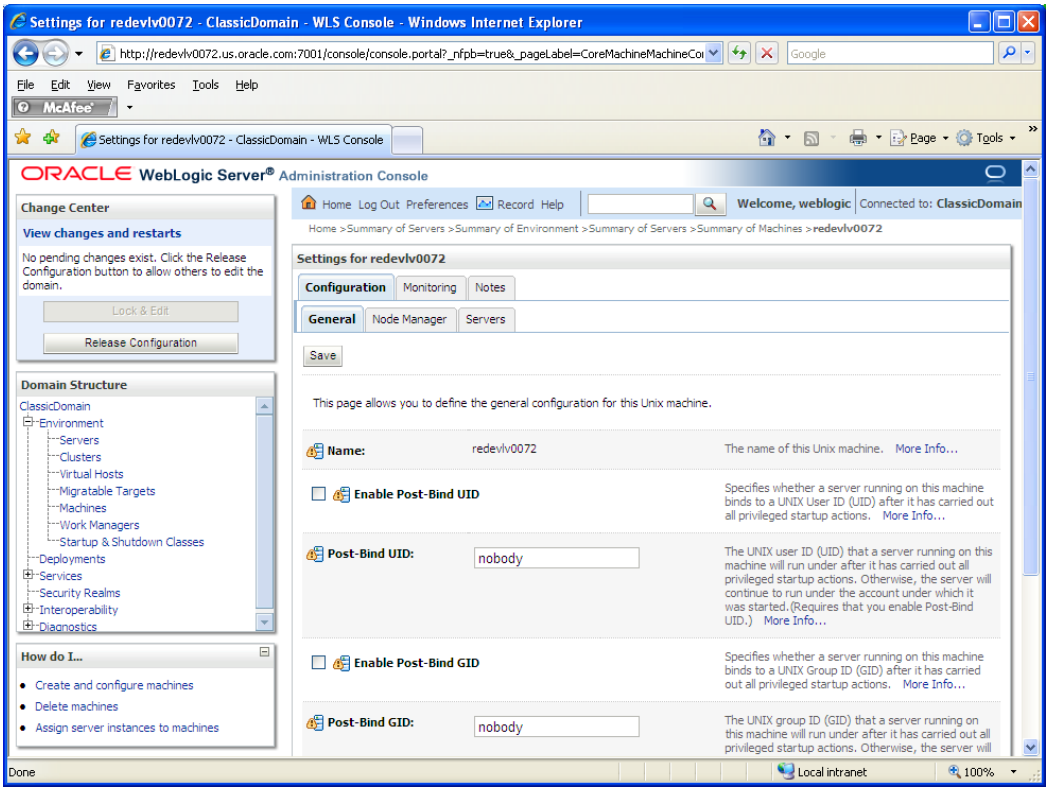
   - Name: Logical machine name
   - Machine OS: UNIX



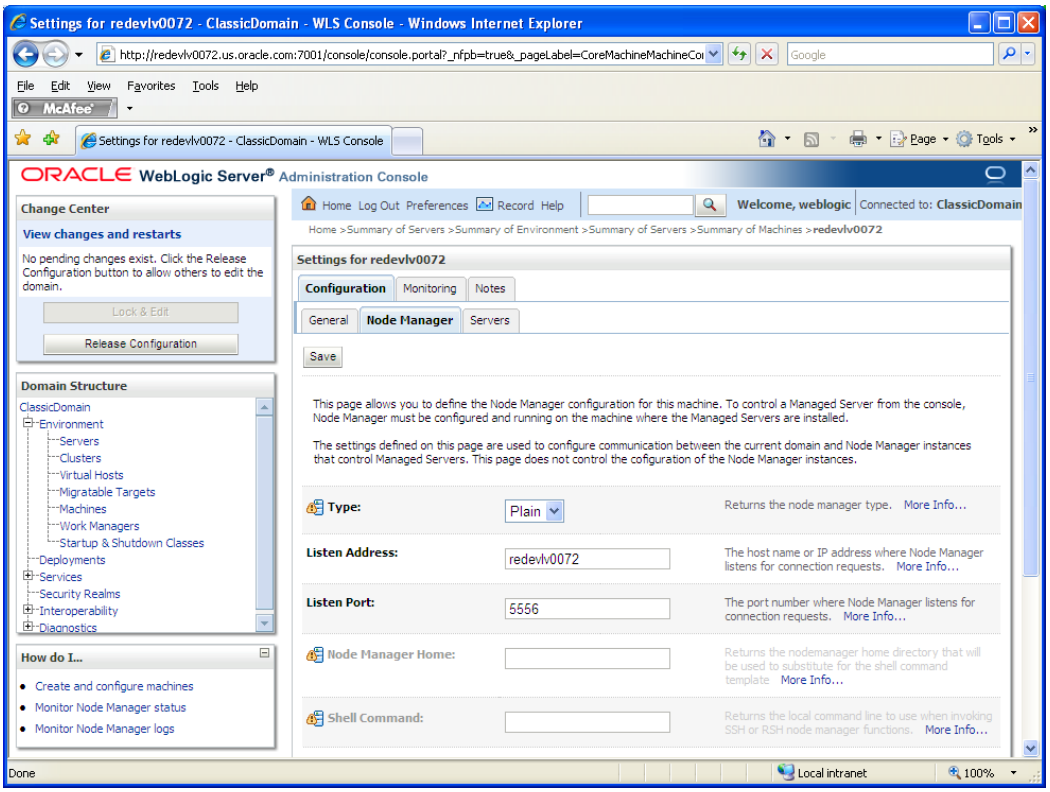
3. Click **OK** to activate the changes.

4. Click the machine created.



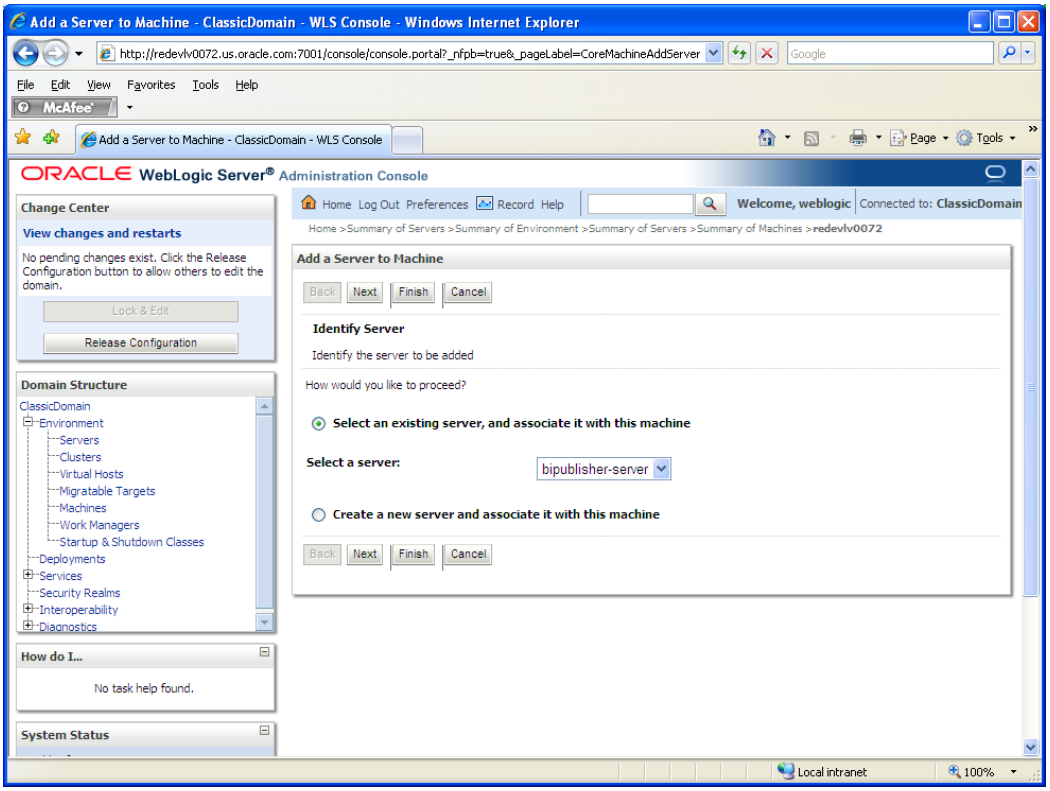


5. Click the NodeManager tab and update the details below.
   - **Type:** Plain
   - **Listen Address**: <weblogic server> (for example, redevlv0072.us.oracle.com)
   - **Listen Port**: Assign a port number. Default port is 5556.



6. Click **Save**.
7. Click **Activate Changes**.
8. Click **Lock & Edit**.

9. Navigate to Environments > machines. Click the machine name. Select the Servers tab. Click **Add**.



10. Add the managed servers that need to be configured with the Nodemanager. Save changes.
    - From the drop down select the managed server to be added to nodemanager
    - Server: <app-server> (for example: bipublisher-server)
11. Click **Next**. Click **Finish**.
12. Click **Activate Changes**.

Note: To activate changes the server needs to be stopped:

<WLS_HOME>/user_projects/domains/<domain_name>/bin/
stopManagedWebLogic.sh bipublisher-server ${server_name}:${server_port}

Go to the managed server that is being added to the machine and click the Server Start tab. In the Class Path box, add the following:

`<full-path-to-domain>/servers/<managed-server>`

For example: `/u00/webadmin/product/10.3.x/WLS/user_projects/domains/<Domain_name>/servers/bipublisher-server`

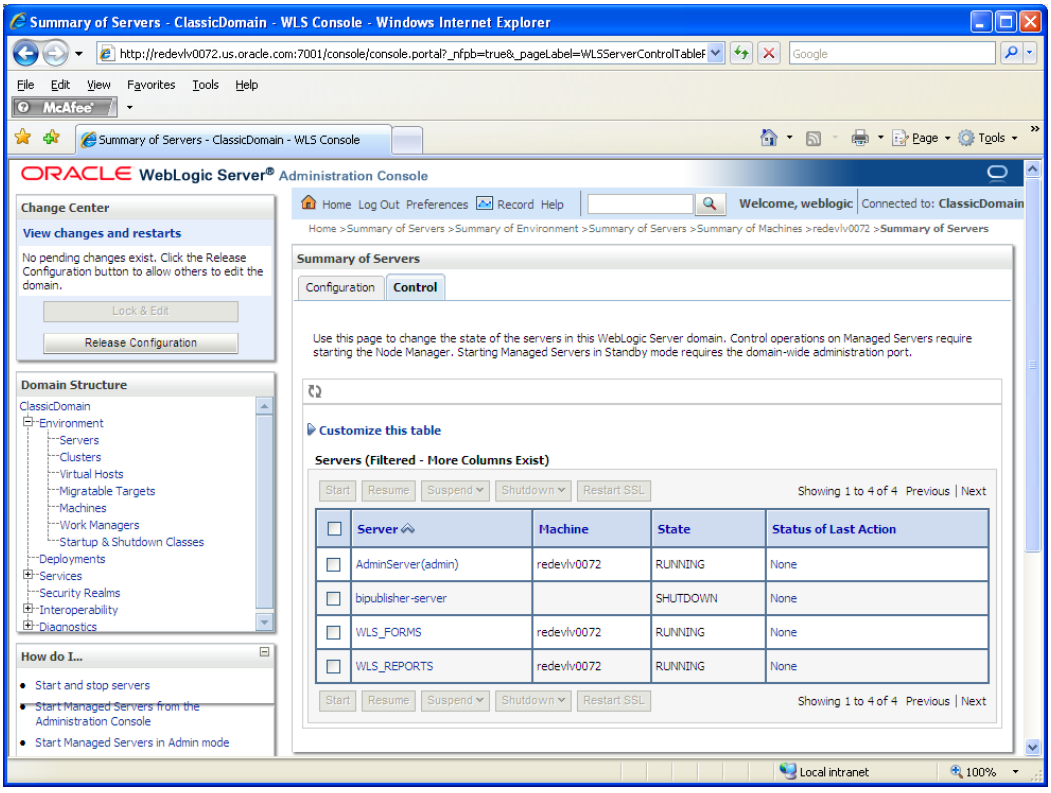
---

13. Click **Save**.
14. Click **Activate Changes**.
15. Edit the nodemanager.properties file at the following location with the below values:
    $WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties
    - SecureListener=false
    - StartScriptEnabled=true
    - StartScriptName=startWebLogic.sh.
16. Start NodeManager from the server using the startNodeManager.sh at <WEBLOGIC_HOME>/wlserver_10.3/server/bin

Start the Managed Servers

To start the managed servers, complete the following steps.

1. Start the Node Manager from the command line if it is not started already.
   `$WLS_HOME/wlserver_10.3/server/bin/startNodeManager.sh`
   After the Node Manager is started, the managed servers can be started through the administration console.
2. Navigate to Environments > Servers. Select <bipublisher-server> managed server. Click the Control tab.



3. Click **Start** to start the managed server.

Additional Setup Steps before Deploying the BI Publisher Application

Following steps are the additional set up steps required before deploying the BI Publisher application in WebLogic.

1. Shut down the bipublisher managed server created above.
2. Add the following option to the startWebLogic.sh script for the server on which the BI Publisher instance is installed.

> **Note:** If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semicolon.

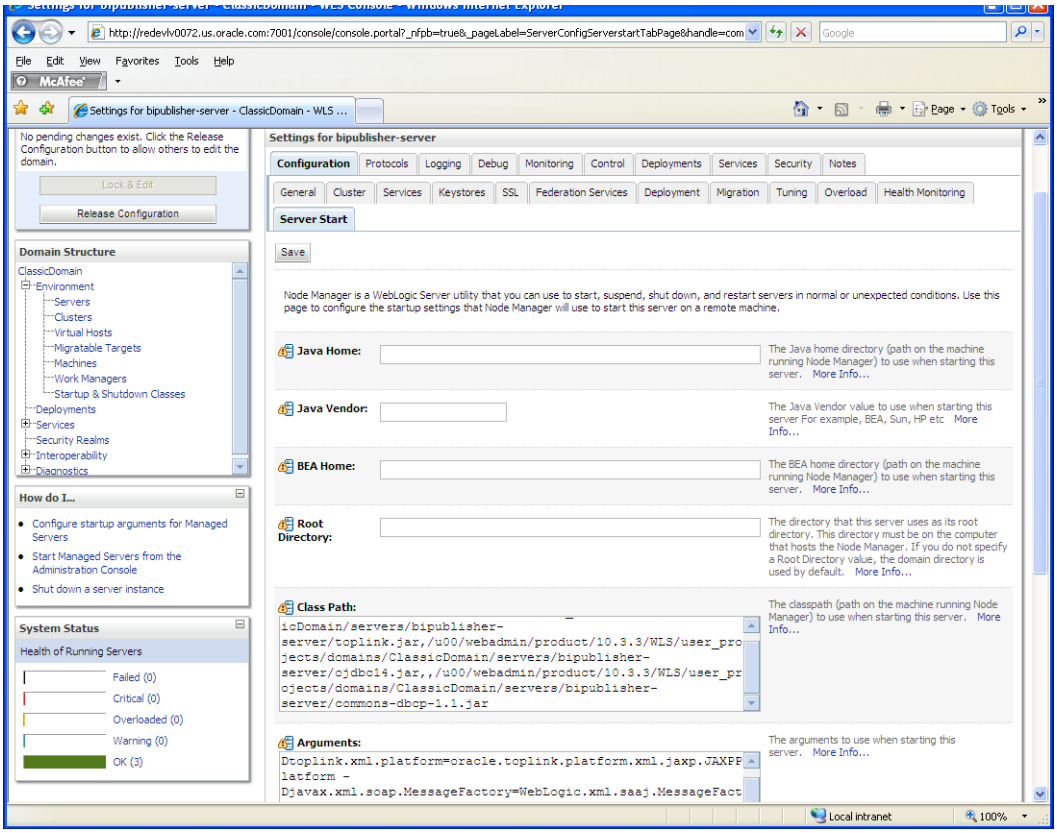
```
JAVA_OPTIONS="${JAVA_OPTIONS} -
Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform"
```

3. Locate the below mentioned jar files in <BI_DEPLOYMENT>/xmlpserver/WEB-INF/lib and add the libraries in your installation (example: copy the jars to the location <WLS_HOME/user_projects/domains/<domain_home>/servers/bipublisher-server/). Append the path of the libraries to the Java classpath for the bipublisher managed server in the weblogic admin console (Classpath in weblogic admin console is available in the path: Weblogic Administration console->Servers->BIPublisher managed server->Server Start-> Classpath)
   - ojdbc14.jar
   - bijdbc14.jar
   - toplink.jar
   - commons-dbcp-1.1.jar
4. Add the following arguments to the arguments of the Java launcher. (Arguments in WebLogic administration console are available through this path: Servers->BIPublisher managed server->Server Start-> Arguments.)

> **Note:** If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semicolon.

```
-Xms512m -Xmx512m -
Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform -
Djavax.xml.soap.MessageFactory=WebLogic.xml.saaj.MessageFactoryImpl
```

> **Note:** The information in the Class Path field in the following screens is for demonstration purposes only. Therefore, version numbers may not reflect the latest version available.



5. Restart the WebLogic server.

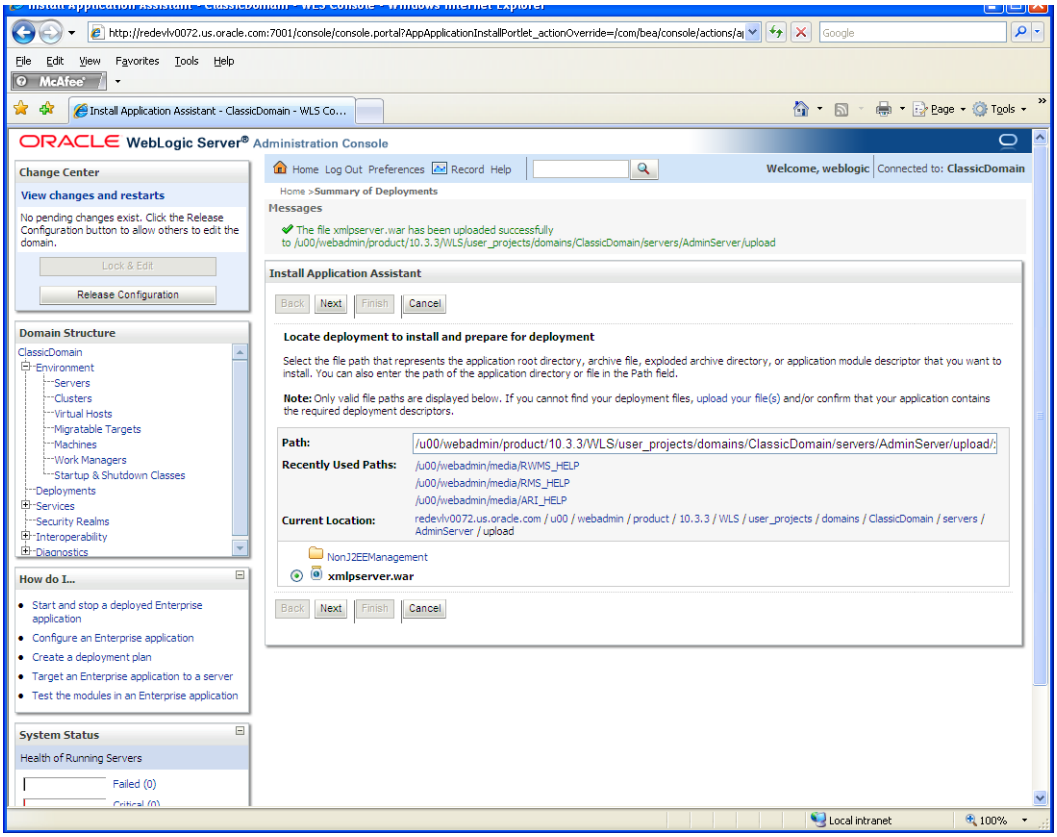
Deploy the BI Application in WebLogic

The exploded archive directory created with the steps in the section, Create an Exploded Directory for the Installation," must be deployed into the bipublisher managed server of WebLogic. Using the WebLogic Administration Console, complete the steps below.

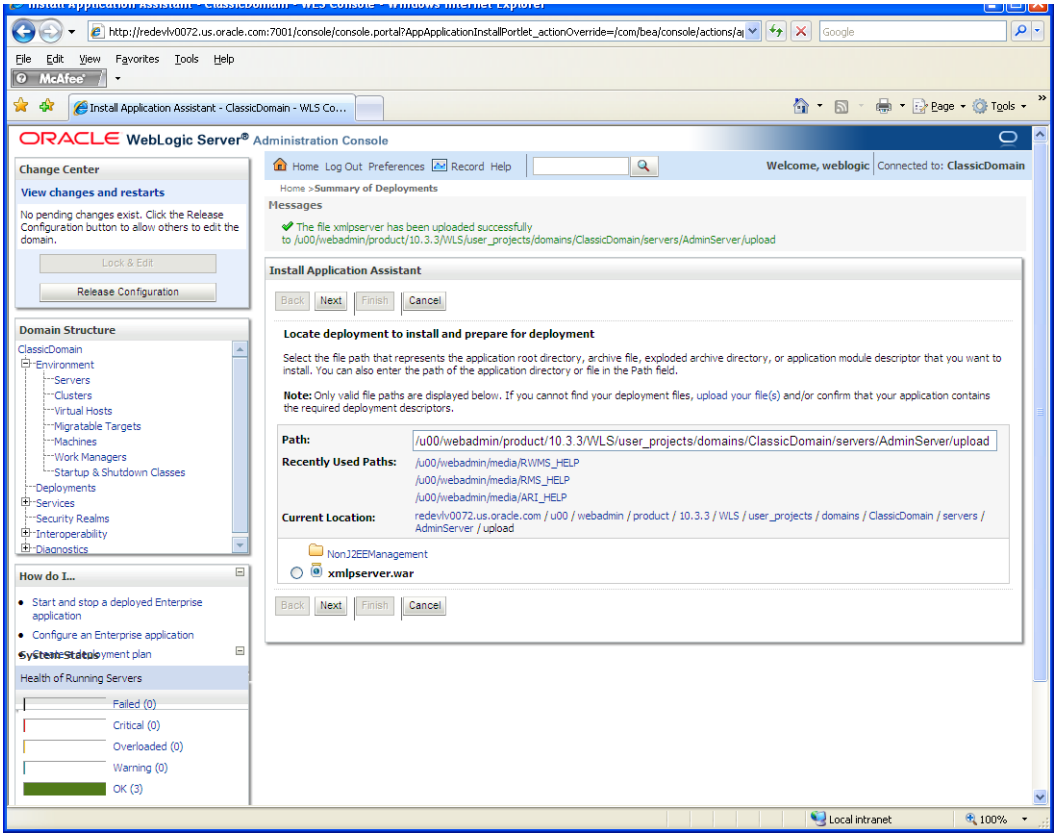
Open the WebLogic Administration Console Web page by entering the appropriate URL for the WebLogic administration server.

For example,
`http://redevlv0072.us.oracle.com:7001/console`

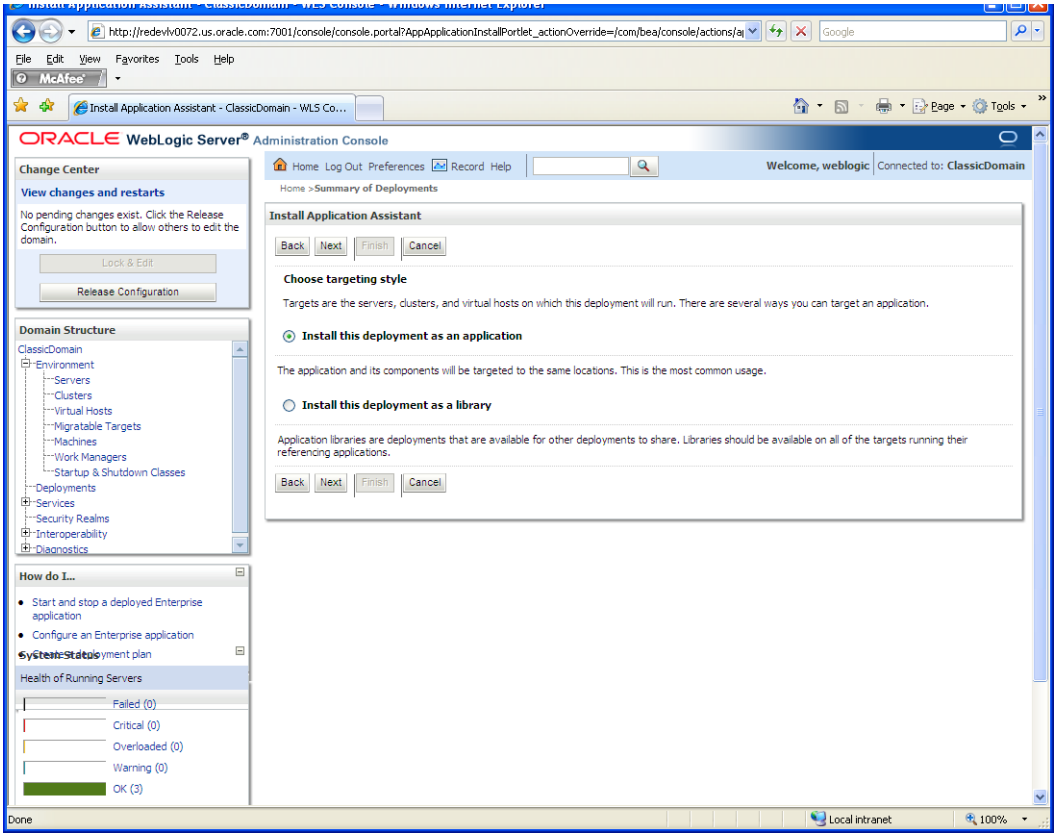
1. Log on to the console using an administrator user name and password.
2. In the Change Center of the Administration Console, click **Lock & Edit**.
3. In the left pane of the Administration Console, click **Deployments**.
4. In the right pane, click **Install**. The following screen should be displayed.



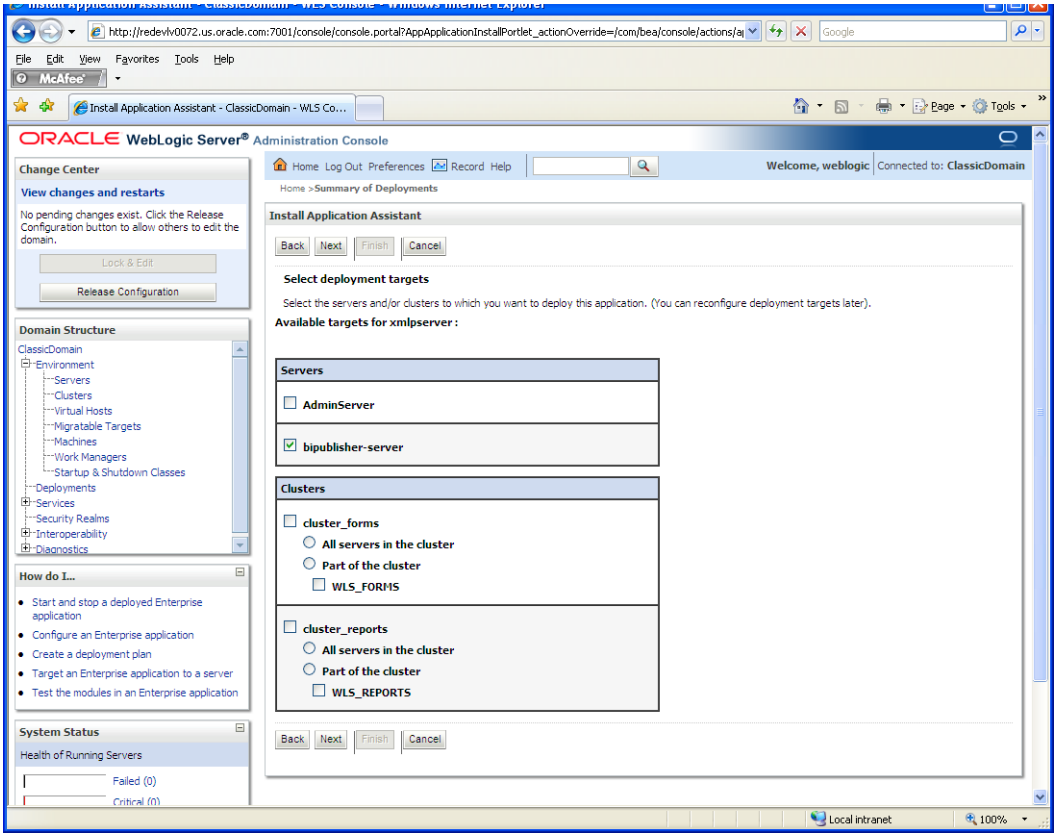
5. Select the exploded directory referenced by <BI_DEPLOYMENT>/xmlpserver. Click **Next**.



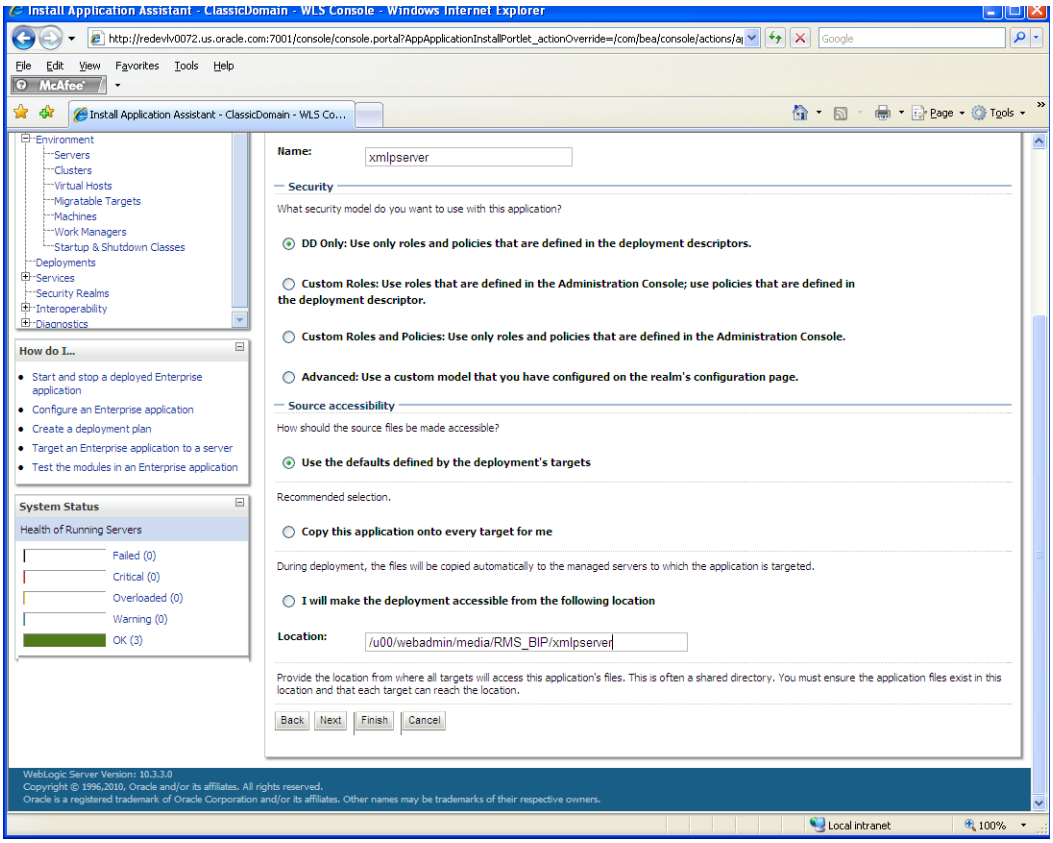
6. Select the **Install this deployment as an application** option. Click **Next**.



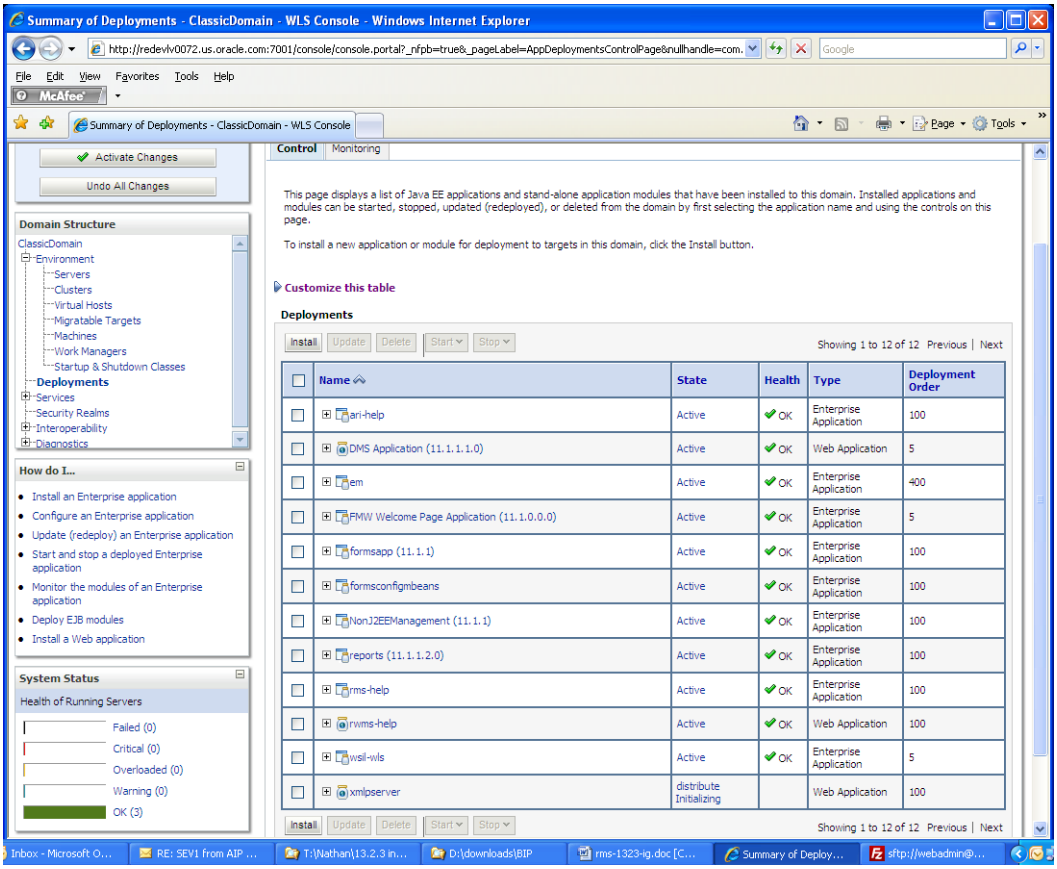
7. Select the bipublisher managed server as the deployment target. Click **Next**.



8. Select **I will make the deployment accessible from the following location** from Source accessibility.

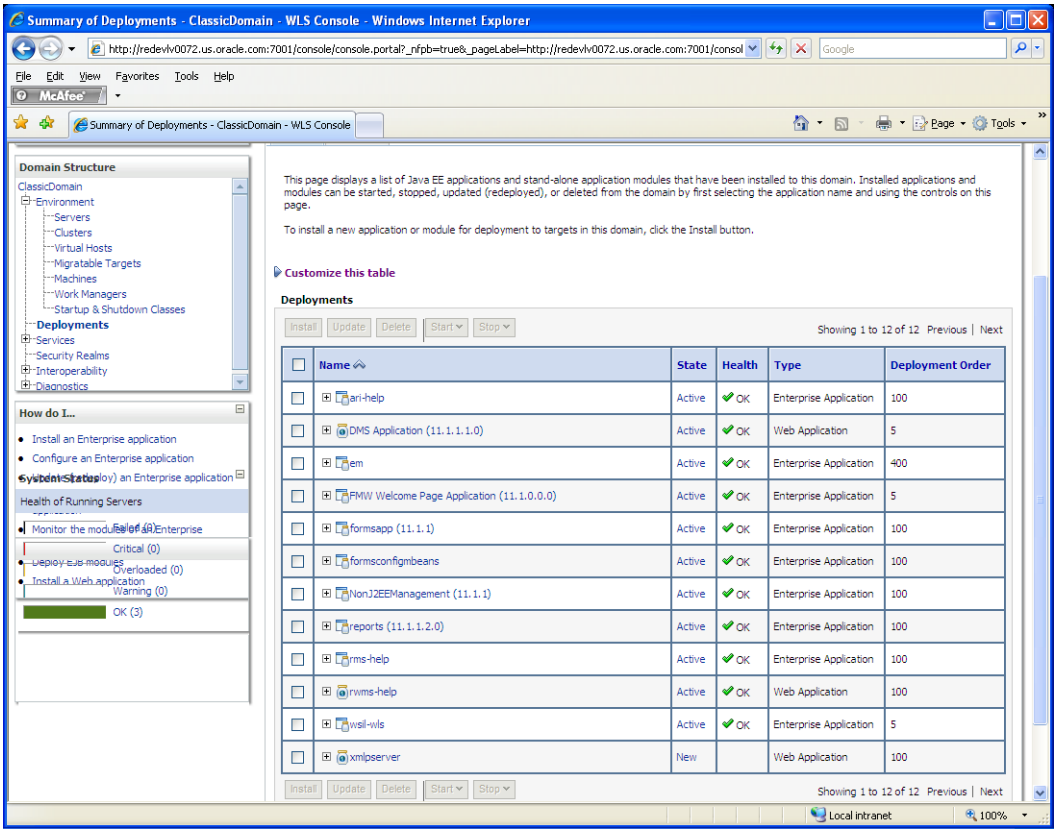


9. Click **Finish**.

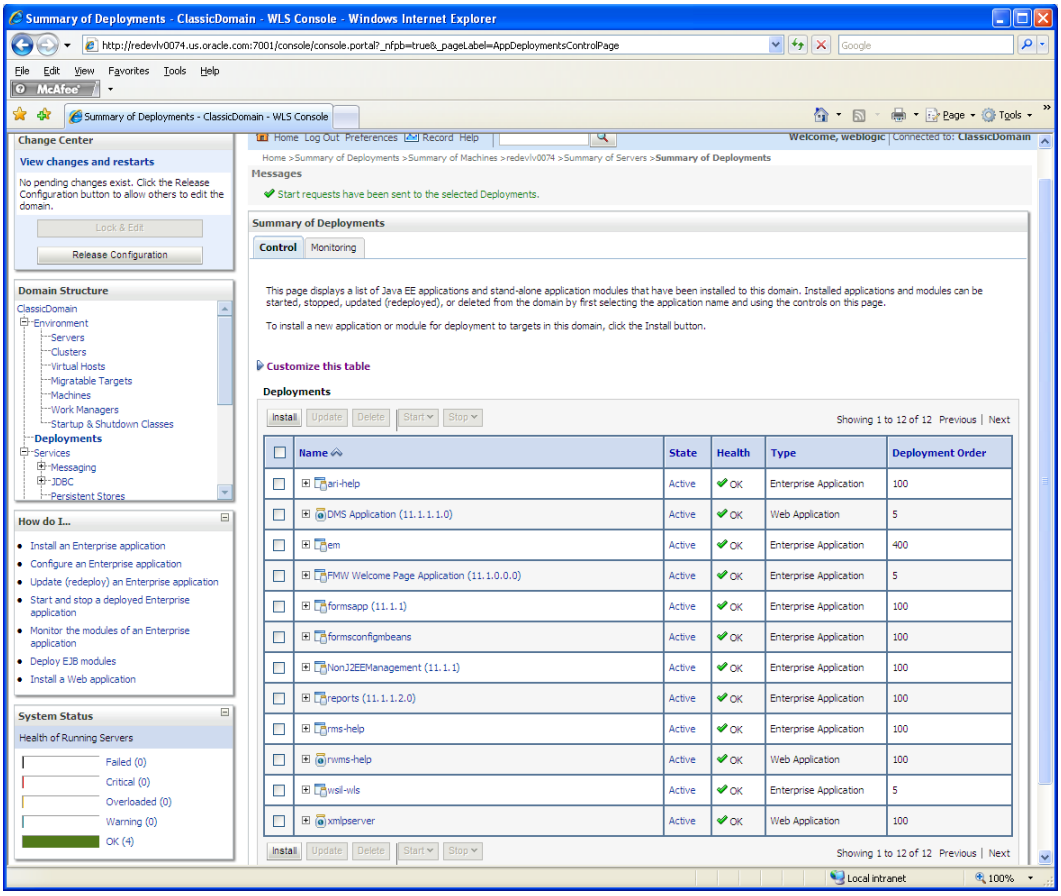


10. To activate these changes, in the **Change Center** of the Administration Console, click **Activate Changes**.

11. On the left side of the console, select Deployments. A list of deployments should appear in the right side of the table.



12. Scroll down and select **xmlpserver** to start the service. The server state should change to active when refreshed.



13. Launch BI Publisher using the appropriate URL for the WebLogic server, where the Web application context is "/xmlpserver".

    For example:

    `http://redevlv0072.us.oracle.com:7001/console`

> **Note:** If using SSO/LDAP for BIPublisher, bipublisher managed server should be started using the following parameter:
>
> -Dweblogic.http.enableRemoteUserHeader=true
>
> If the <bipublisher-server> is being started from WebLogic Administration Console, the above parameter should be added here before restarting the bipublisher managed server from the Administration Console:
>
> Adminconsole > Environment > Servers-> <bipublisher-server> > Server Start > Arguments.
>
> If the server is being started from UNIX, consider the following example:
>
> <WEBLOGIC_DOMAIN_HOME>/bin/startManagedWeblogic.sh <bipublisher-server> <server:port> -Dweblogic.http.enableRemoteUserHeader=true

Manually Copy Reports to Install Directory

If you followed the instructions under "Option 1: Use Application Installer to Patch" in the Chapter 5, "Patch Application Server Installation Tasks", you can skip to the next section ("Installing the RWMS BI Publisher Templates"). If you followed "Option 2: Compile RWMS Forms Directly," you must manually copy the reports and related files to INSTALL_DIR.

1. If INSTALL_DIR/base/reports/10g already exists skip to step 2. Otherwise, follow these steps:
   a. Change directories to INSTALL_DIR/base/
   b. Make a new directory called "10g"
   c. Change directories to INSTALL_DIR/base/10g
   d. Make new directories called "RWMS REPOSITORY" and "RWMS SYSTEM REPOSITORY"
   e. Move the Reports directories from INSTALL_DIR/base/reports to the INSTALL_DIR/base/10g/RWMS REPOSITORY and INSTALL_DIR/base/10g/RWMS SYSTEM REPOSITORY directories, based on the following table. In the table below, check the Mixed mode representation. The reports with Mixed mode set to Y must be copied to the RWMS REPOSITORY and RWMS SYSTEM REPOSITORY directories. The reports with Mixed mode set to N must be copied to only the RWMS SYSTEM REPOSITORY directory.

Reports(New File Name)	Mixed
asn_recv_package_audit_sys	Y
asn_receiving_receipt_sys	Y
best_before_date	N
bill_of_lading_sys	Y
container_contents_lbl	N
container_manifest_sys	Y
gift_card_report_sys	N
inventory_by_item	N
inventory_by_location	N
pack_slip_labels_sys	N
pick_package_audit_sys	Y
pts_containers_to_close_sys	N
quality_audit_sys	Y
receiving_receipt_sys	Y
receiving_register	N
recv_package_audit_list_sys	Y
return_to_vendor	N
rtv_advice_sys	Y

Reports(New File Name)	Mixed
unit_pick_group_sys	N
unresolved_appointment	N
wave_preview_sys	N
unit_pick_labels_lbl	N
error_log	N
cycle_count_log	N
confirm_fwd_case_picks_sys	N
qualified_container_sys	Y
pick_labels_lbl	N
receiving_labels_lbl	N
ship_labels_lbl	N
ticketing_lbl	N
generic_lbl	N

f. INSTALL_DIR/base/reports should now be empty. Move INSTALL_DIR/base/10g/ to INSTALL_DIR/base/reports/10g

2. Copy the reports and extras directories from RWMS application patch APP_PATCH_DIR/app-patch/<version>/ to the reports and extras directories under INSTALL_DIR/base/. This step should be done with each version in order of earliest to latest patch starting at 13.2.3.

BiPublisher 10g – Installing RWMS BI Publisher Templates

This section describes how the RWMS report templates are installed into the appropriate BI Publisher server repositories. BI_REPOSITORY refers to the BI Publisher reports repository.

Example: /u00/webadmin/RWMS_BIP/xmlpserver/XMLP

Report files are placed in the directory, INSTALL_DIR/base/reports, and must be copied into the BI repository directory.

3. If necessary, create a <BI_REPOSITORY>/Reports/Guest/RWMS13 folder in the BIPublisher repository.
4. Change directories to INSTALL_DIR/base/reports/10g. This directory contains subdirectories whose names reflect the names of report templates provided with RWMS.
5. Copy each report directory into the directory created above

   Example,

   ```
   cp -R *  /u00/webadmin/RMS_BIP/xmlpserver/XMLP/Reports/Guest/RWMS13/
   ```

6. Set up the following JAR files.

   In the .env file, the CLASSPATH variable was set to look for the JAR files listed below. If the directory does not already exist as specified in the .env file create it and place the jar files in there from these locations:

   - INSTALL_DIR/base/retail-public-security-api/lib/retail-public-security-api.jar

- INSTALL_DIR/base/extras/wmsSecurity.jar
- INSTALL_DIR/base/extras/bihelper/bihelper.jar – extract this JAR file.

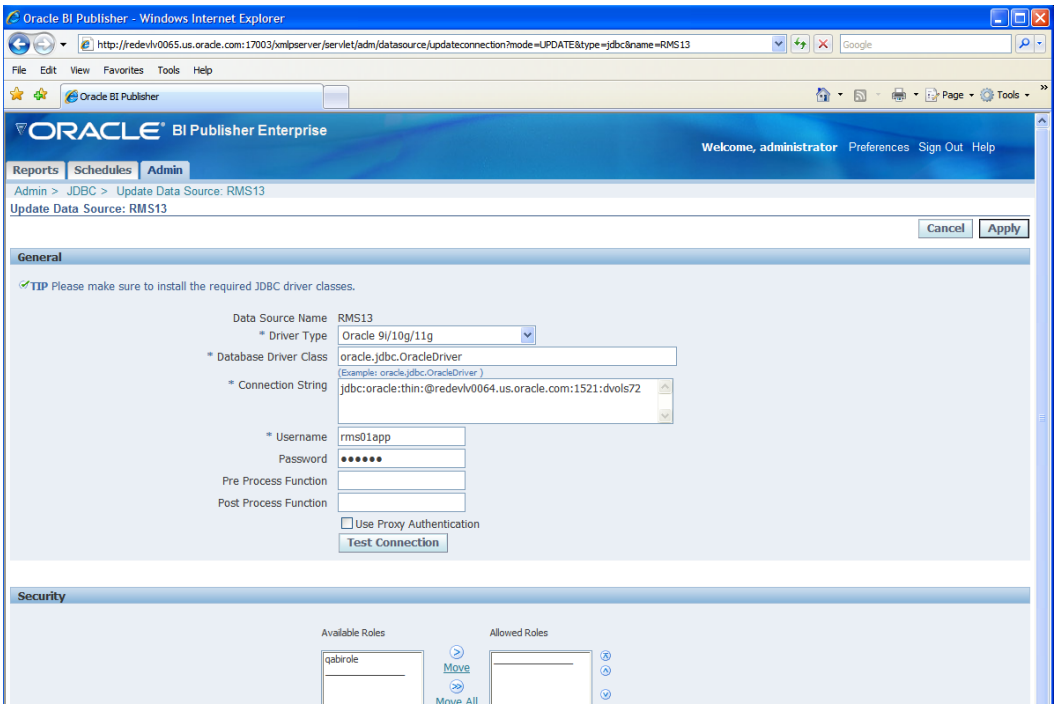
> **Note:** The rt.jar and jacob.jar JAR files are located on the application server but they are not specified in the CLASSPATH variable inherently. As a part of the .env file configuration you should have added them to the CLASSPATH. Make sure you add the JAR files to the directory specified in the CLASSPATH.
>
> You must use a find –name .<filename> command to locate them on the server.

7. Verify that you have configured the following BI Publisher configuration points, as indicated in BI Publisher documentation:
   a. Create an administrative user. The user name and password must be the same as the user created during the installation process for the Java wallet. If you are not using the Java wallet, the user name and password must match the values in the.emv file for RWMS_BI_USER and RWMS_BI_PWD.
   b. Configure printers, including Label and LAN printers.

Configuring the RWMS JDBC connection

Complete the following steps to configure JDBC connection for RWMS Data Source name. This is the data source that RWMS uses 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 and select the **JDBC Connection** hyperlink in the Data Sources lists. The following screen is displayed.



3. Enter the appropriate details for the RWMS data source. Once the data is entered, click Test Connection to test the connection. Driver type is ORACLE 11g and 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>

> **Note:** The reports require the connection name to be set as RWMS.

Configure the BIPublisher Scheduler

Complete the following tasks to configure the scheduler:

1. Create the database user for scheduler configuration as below:

```
create user <scheduler schema user> identified by <password> default
tablespace <table space name> temporary tablespace temp;
grant create session,create table to <scheduler schema user>;
alter user <scheduler schema user> quota unlimited on <table space name>;
```

2. Navigate to the top level Admin display and select the Scheduler Configuration hyperlink. This will show you the schedule configuration screen which is very similar to the JDBC connection screen in the previous instruction topic Enter the appropriate database connection details and test the connection as in the previous steps. If this connection operates successfully, save the connection details. Click **Install Schema** to install the schema for BI Publisher using the <scheduler schema user> created in the previous step.

> **Note:** For information about configuring BI Publisher, use the following URL:
>
> http://download.oracle.com/docs/cd/E12844_01/doc/bip.1013/e12690/T434820T487783.htm#5187634

Verify Oracle BI Publisher Set Up for RWMS Reports

Verify that Oracle BI Publisher has been set up correctly as follows:

1. Click the **Admin** tab. Under System Maintenance, click **Report Repository**. The Path variable should be set as part of the BI Publisher installation, REPORTS_DIR.
2. Update the file ORACLE_HOME/forms/server/rwms.env with the following variables:
   - RWMS_REPORTS_TEMP – $ORACLE_HOME/user_projects/domains/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_11.1.2/e18uoi/war/rwms/reptemp/.

     > **Note:** create reptemp folder in the above directory location.

   - RWMS_REPTEMP_ALIAS – /forms/rwms/reptemp/
   - RWMS_REPORTS_URL – http://server:7001/xmlpserver/servlet/report?f=/Guest/RWMS/RWMS+REPOSITORY
     - Where 7001 is the port where xmlpserver has been deployed.
   - RWMS_BI_USER – Administrator user name.
   - RWMS_BI_PWD – Associated administrator password.
   - RWMS_SYS_REPORTS_DIR – /Guest/RWMS/RWMS SYSTEM REPOSITORY/
   - RWMS_END_POINT_URL – http://server:7001/xmlpserver/services/PublicReportService?wsdl
     - Where 7001 is the port where xmlpserver has been deployed.

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 for 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. The following is a checklist after the section on "Configuring Forms to use BI and WebUtil" is the next section are complete :

- Verify you executed the create_webutil_db.sql
- Verify you moved jacob.jar to forms/java folder and also signed it.
- Verify you modified the <INSTALLATION_NAME>.env file to correct jacob.jar location in CLASSPATH.
- Verify you compiled webutil.pll
- Verify you edited the webutil.cfg file and set transfer.database.enabled=TRUE

Configuring Forms to use BI and WebUtil

This section provided 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


[rwms13inst]
envfile=./rwms13inst/rwms13inst.env
width=950
height=685
separateFrame=true
lookAndFeel=Oracle
colorScheme=swan
archive=frmall.jar, icons.jar,frmwebutil.jar,jacob.jar
form=logon_scr.fmx
userid=/@rwms13inst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwms13inst_hh]
envfile=./rwms13inst/rwms13inst.env
width=100%
height=100%
separateFrame=false
form=hh_intro_s.fmx
lookAndFeel=Oracle
Logo=false
colorScheme=swan
archive=frmall.jar,icons.jar
imageBase=codebase
```

```
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/r
wms13inst/fmrweb.res usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0 marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=/@rwms13inst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwms13inst_tm]
envfile=./rwms13inst/rwms13inst.env
width=100%
height=100%
separateFrame=false
form=tm_intro_s.fmx
Logo=false
lookAndFeel=Oracle
archive=frmall.jar,icons.jar
imageBase=codebase
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/r
wms13inst/fmrweb.res usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0 marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
userid=/@rwms13inst
#ssoMode=true
#ssoDynamicresourceCreate=true

[rwms13inst_wr]
envfile=./rwms13inst/rwms13inst.env
width=100%
height=100%
separateFrame=false
form=wr_intro_s.fmx
Logo=false
background=false
lookAndFeel=Oracle
archive=frmall.jar,icons.jar
imageBase=codebase
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/r
wms13inst/fmrweb.res usesdi=YES
splashScreen=false
HTMLbodyAttrs=scroll="no" topmargin="0" leftmargin="0 marginheight="0"
marginwidth="0" onload="window.moveTo(0,0);"
```

```
userid=/@rwms13inst
#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/rwms13inst/
```

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.x/WLS_Forms/user_projects/domains/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_11.1.2/e18uoi/war/rwms/reptemp/

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 mandatory 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

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.

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.

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

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	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/

Field Title	CLASSPATH
Field Description	You will need to add the jars listed below: **jacob.jar** For some reason this jar is not already in the classpath on some servers – To make sure this is not an issue go find it on the server and put it in the extras directory. **bihelper.jar** This Jar needs to be extracted in the directory it is located in. You may also have to physically go get this file and put it in here. NOTE: Put it in the extras directory. The **wmsSecurity.jar** is already being placed there by the installer.

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

```
…
CLASSPATH=/u00/webadmin/product/10.3.x/WLS_Forms/as_1/forms/j2ee/frmsrv.j
ar:/u00/webadmin/product/10.3.x/WLS_Forms/as_1/jlib/ldapjclnt11.jar:/u00/
webadmin/product/10.3.x/WLS_Forms/as_1/jlib/debugger.jar:/u00/webadmin/pr
oduct/10.3.x/WLS_Forms/as_1/jlib/ewt3.jar:/u00/webadmin/product/10.3.x/WL
S_Forms/as_1/jlib/share.jar:/u00/webadmin/product/10.3.x/WLS_Forms/as_1/j
lib/utj.jar:/u00/webadmin/product/10.3.x/WLS_Forms/as_1/jlib/zrclient.jar
:/u00/webadmin/product/10.3.x/WLS_Forms/as_1/reports/jlib/rwrun.jar:/u00/
webadmin/product/10.3.x/WLS_Forms/as_1/forms/java/frmwebutil.jar:/u00/web
admin/product/10.3.x/WLS_Forms/as_1/jlib/start_dejvm.jar:/u00/webadmin/pr
oduct/10.3.x/WLS_Forms/as_1/opmn/lib/optic.jar:/u00/webadmin/rwmsTESTING/
base/extras/jacob/jacob.jar:/u00/webadmin/rwmsTESTING/base/extras/rt/rt.j
ar:/u00/webadmin/rwmsTESTING/base/extras/bihelper/bihelper.jar
…


…
#RWMS Application Server added lines

NLS_DATE_FORMAT=DD-MON-RR
NLS_LANG=AMERICAN_AMERICA.UTF8

FORMS_USERNAME_CASESENSITIVE=1
FORMS_REJECT_GO_DISABLED_ITEM=FALSE

ORACLE_RWMS_EXTRAS=/vol.rtk/pkg_mocks/rwms132/install/base/extras

RWMS_REPORTS_URL=http://mspdv351.us.oracle.com:9001/xmlpserver/servlet/re
port?f=/Guest/RWMS/RWMS+REPOSITORY
RWMS_REPORTS_TEMP=/u00/webadmin/RWMS_BIP/xmlpserver/XMLP/ReportsTemp

RWMS_SYS_REPORTS_DIR=/Guest/RWMS/RWMS SYSTEM REPOSITORY/
RWMS_END_POINT_URL=http://mspdv351.us.oracle.com:9001/xmlpserver/services
/PublicReportService?wsdlURL

RWMS_WALLET_PATH=/vol.rtk/pkg_mocks/rwms132/install/base/.javawallet
RWMS_WALLET_PARTITION=rwms13inst
RWMS_WALLET_LOGON=TRUE
RWMS_BI_USER=
RWMS_BI_PWD=BI_ALIAS
RWMS_DB_CONNECT=RWMS132MOCK_pkols07
RWMS_FORMS_SERVER=http://mspdv351:9001
RWMS_REPTEMP_ALIAS=/forms/rwms/reptemp/
RF_LAUNCH_VALUE_ALIAS=/forms/rwms/rf_launch/

CLASSPATH=$CLASSPATH:/u00/webadmin/rwmsTESTING/base/extras/wmsSecurity.ja
r
```

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.x/WLS_Forms/user_projects/domains/ClassicDomain/servers/WLS_FORMS/tmp/_WL_user/formsapp_11.1.2/e18uoi/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 location specified by the rf_launch_value alias in the httpd.conf. 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, and the port specified in the httpd.conf 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";

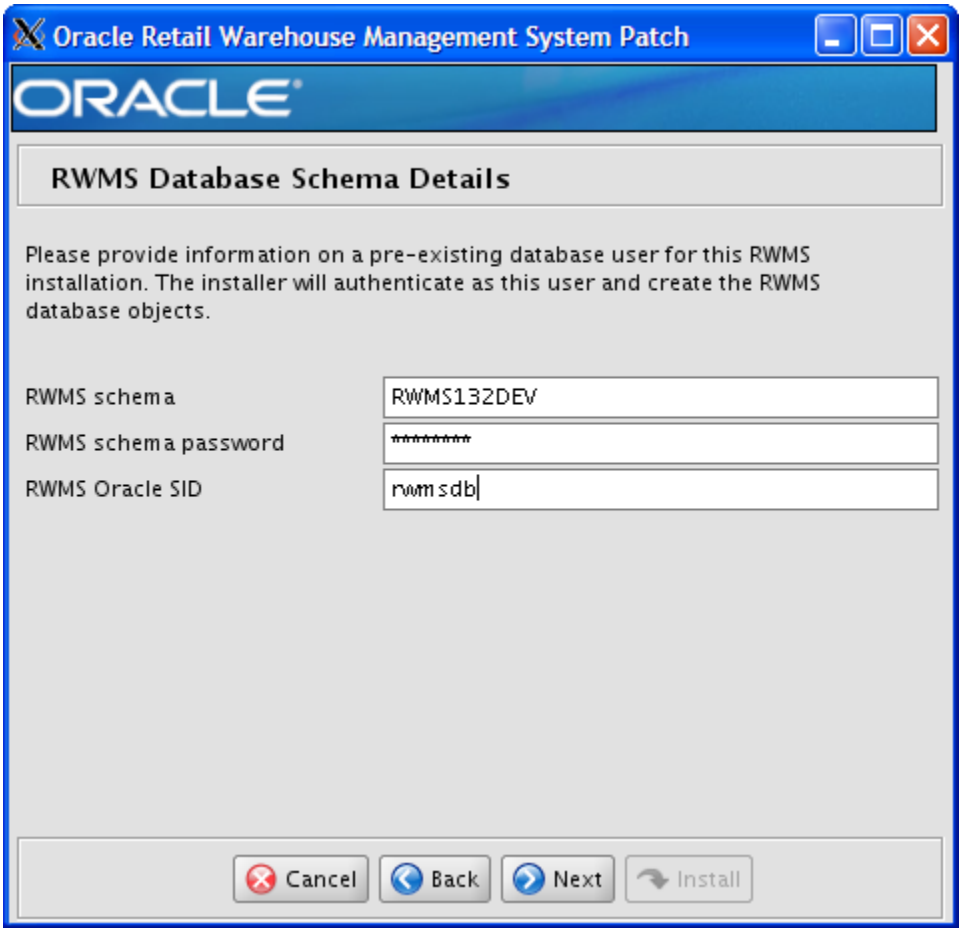
- var wr_device = “http:<Server_Name>:<port>/forms/frmservlet?config=<INSTALLATION_NAME>_wr";
- var exit_script = "http://<Server_Name>:<port>/forms/rwms/rf_launch/close.htm";

A

Appendix: RWMS Database Patch Installer Screens

The following details about your environment are necessary for the installer to successfully patch the RWMS database schema.

Screen: Database Schema Details

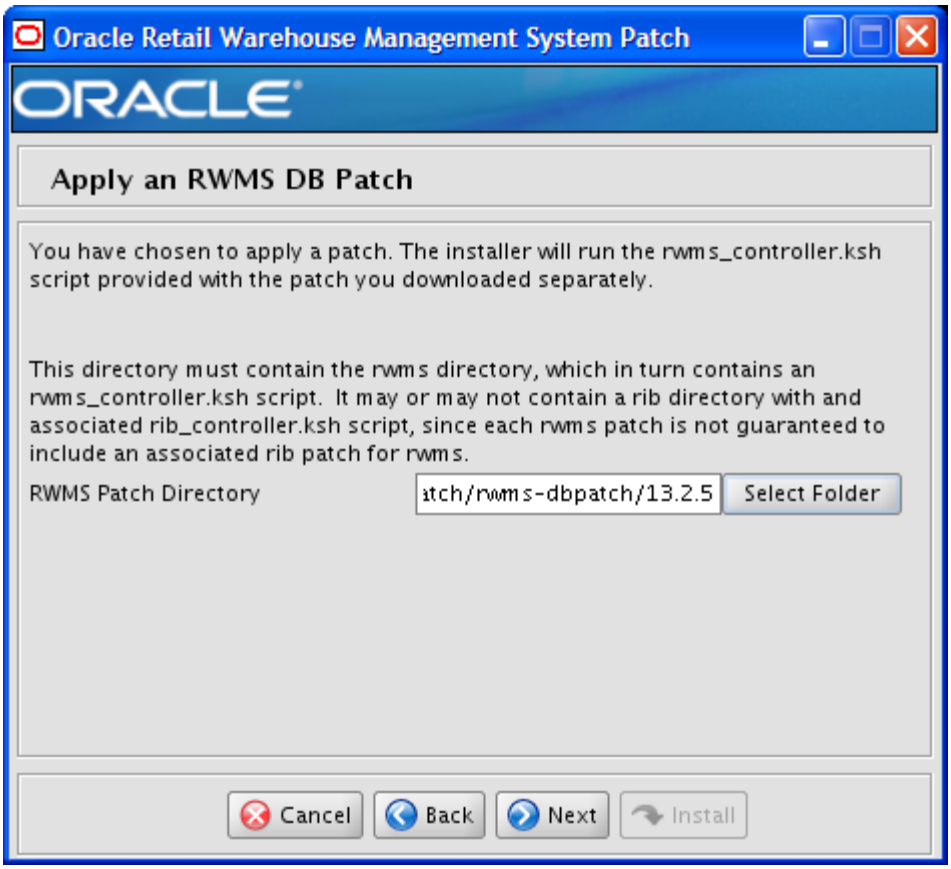


Field Title	RWMS schema
Field Description	Provide the RWMS database user here. The installer logs into the database as this user to create the RWMS schema. This user must already exist in the database when the RWMS database schema installer is run.
Example	RWMS132DEV

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

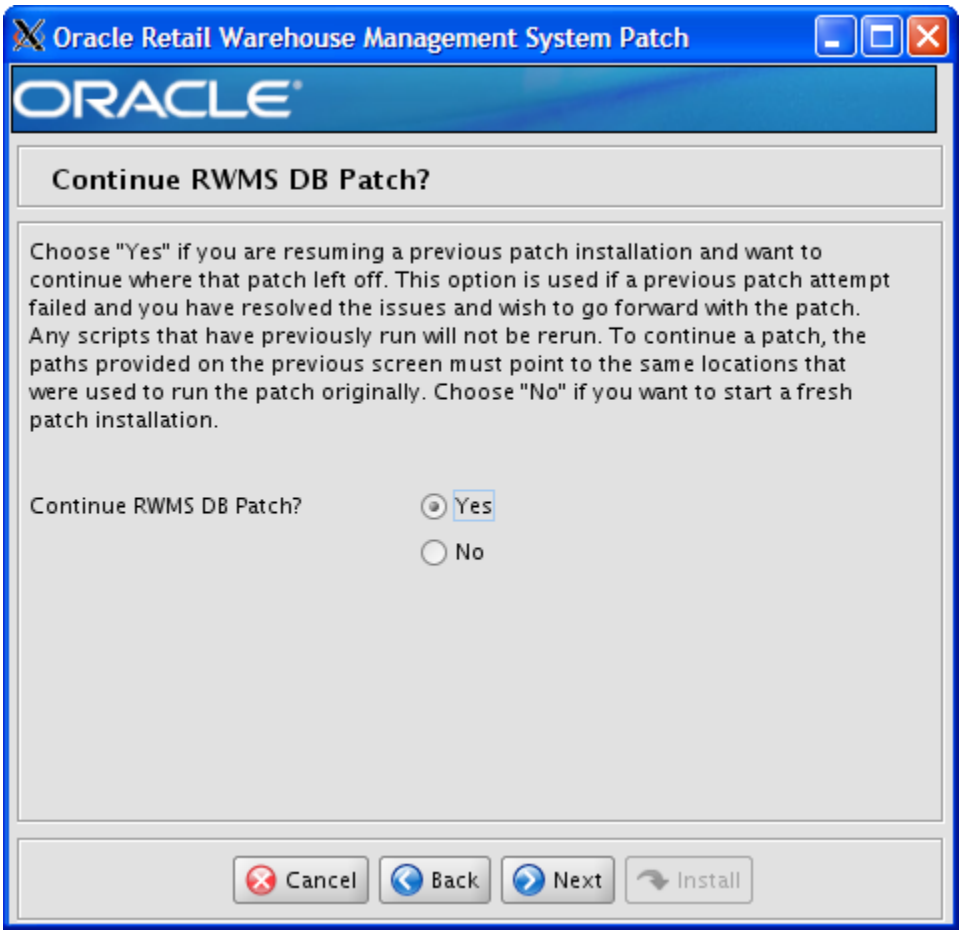
Field Title	RWMS Oracle SID
Field Description	Oracle system identifier for the database where the RWMS patch will be applied.
Example	rwmsdb

Screen: Apply an RWMS DB Patch



Field Title	RWMS Patch Directory
Field Description	Provide the directory path to the RWMS patch you want to install. The installer runs only the patch you provide. **Note:** The directory you choose must contain rwms/rwms_controller.ksh, and may contain rib/rib_controller.ksh.
Example	/path/to/rwms/dbschemapatch/rwms-dbpatch/<version> for all 13.2.x patches **Note:** The patch option is intended for patches starting with 13.2.4.

Screen: Continue RWMS DB Patch



Field Title	Continue RWMS DB Patch?
Field Description	The patch process allows you to continue a previously run patch if it stopped before completion or failed. If Yes is selected, any scripts that were previously run for the RWMS patch will be skipped. If No is selected, the patch will start from the beginning. **Note:** To continue a patch, the content of the "processed" directories in the RWMS Patch chosen on the previous screen must be the same as it was after the previous patch was stopped. If you choose No, this directory will be cleared, and you will not be able to continue this patch in the future.

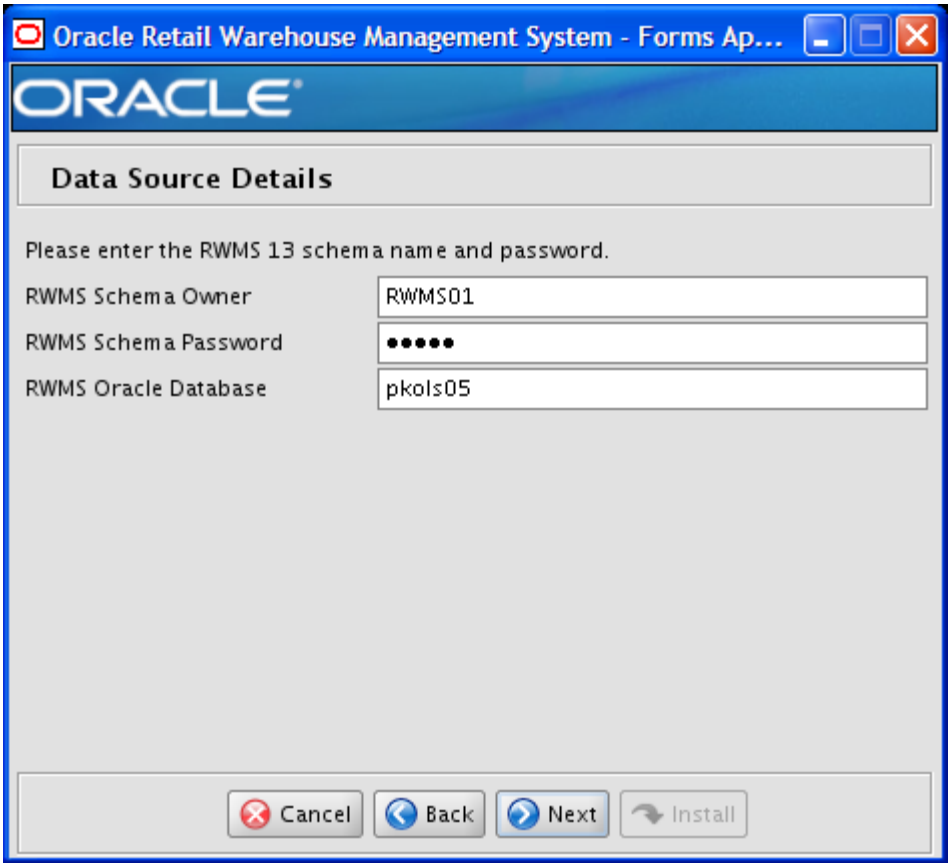
B

Appendix: RWMS Application Installer Screens

Screen: Oracle Customer Information

For information about this screen, see the "Oracle Configuration Manager" section in the *Oracle Configuration Manager Installer Guide*.

Screen: Data Source Details

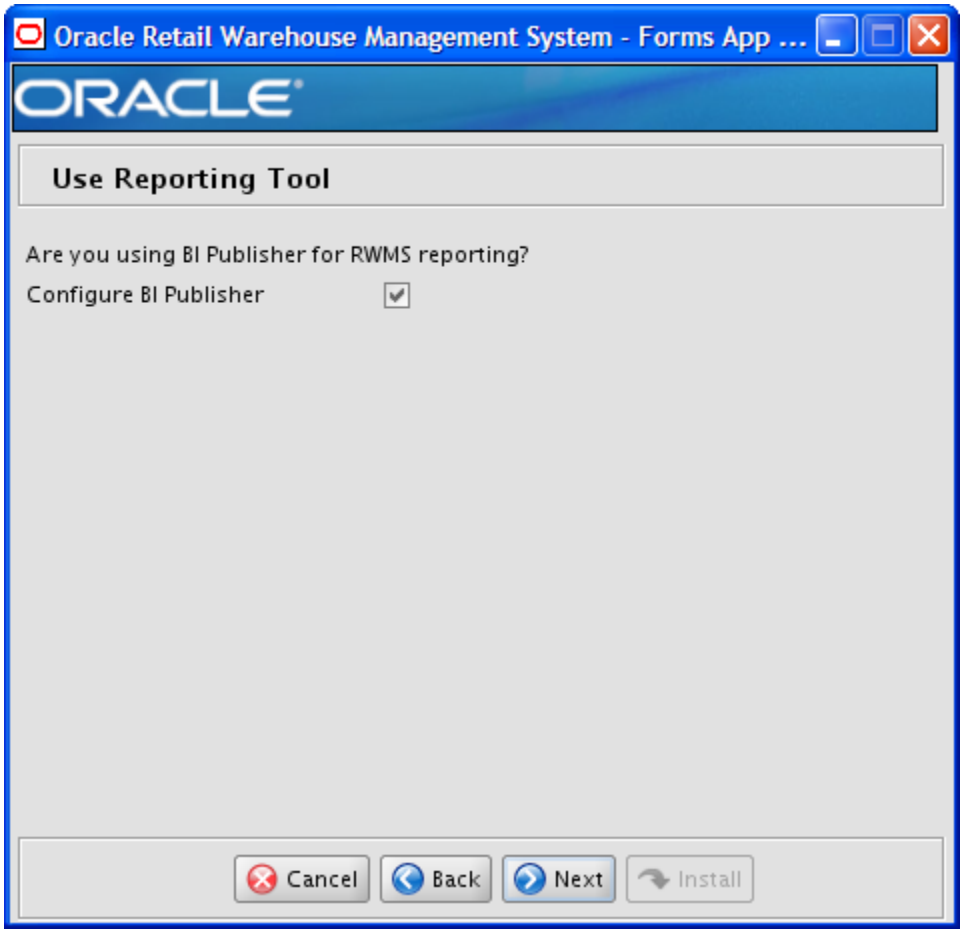


Field Title	RWMS Schema Owner
Field Description	This is the same username that was used during the RWMS Database Schema Installer.
Example	RWMS01

Field Title	RWMS Schema Password
Field Description	This is the same password that was used during the RWMS Database Schema Installer.

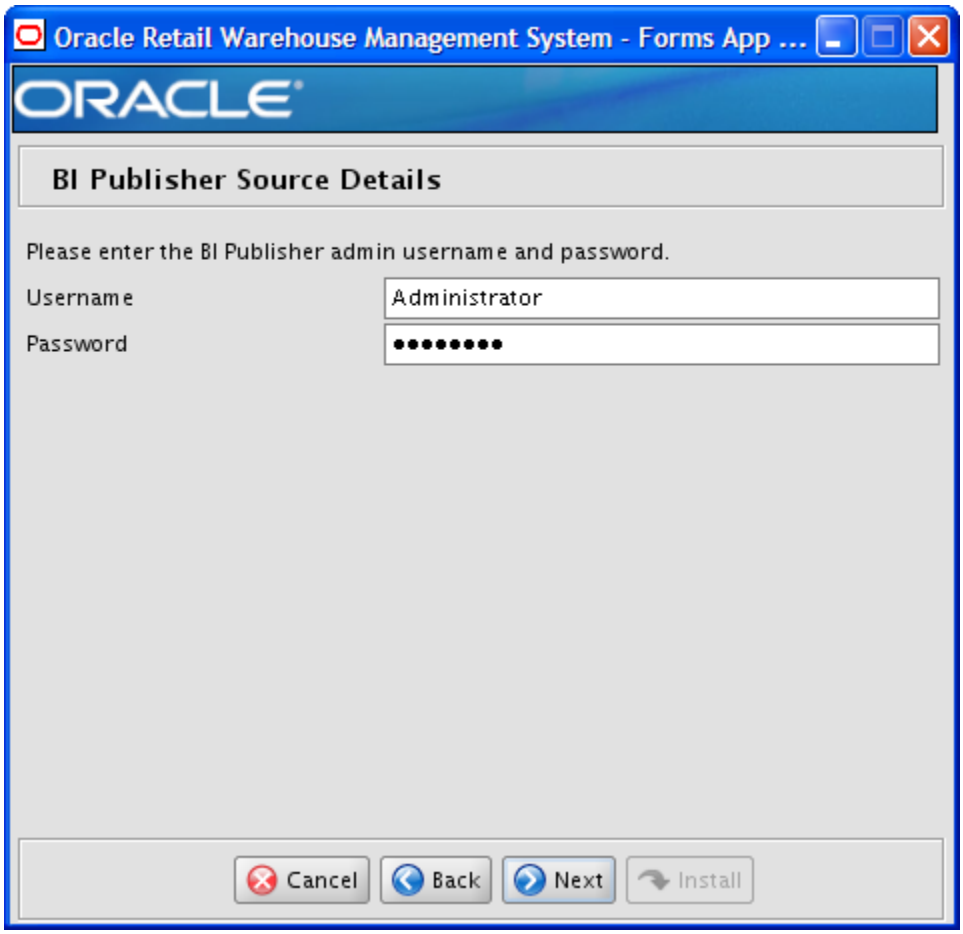
Field Title	RWMS Oracle SID
Field Description	This is the same Oracle SID that was used during the RWMS Database Schema Installer.
Example	pkols05

Screen: Use Reporting Tool



Field Title	Configure BI Publisher
Field Description	Check this box if you are using BI Publisher with RWMS.

Screen: BI Publisher Source Details



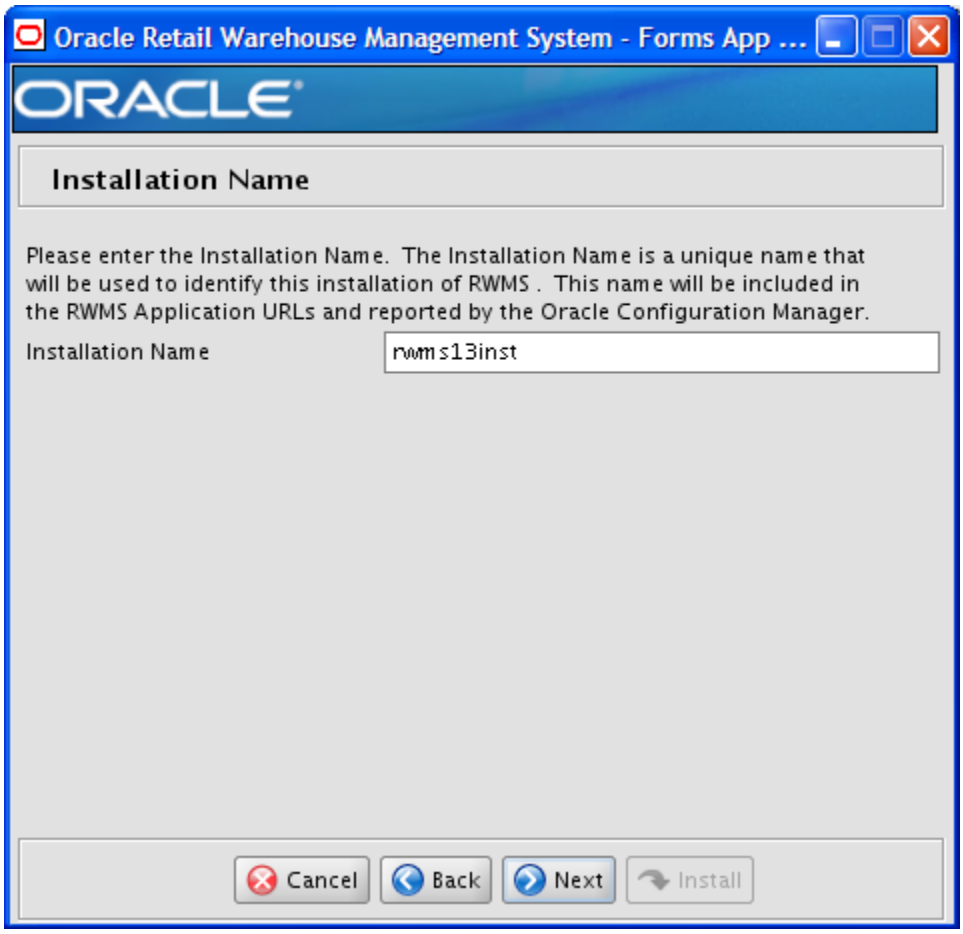
Field Title	Username
Field Description	This is the admin username for BI Publisher.
Example	Administrator

Field Title	Password
Field Description	This is the password for the BI Publisher admin

Screen: Oracle Wallet

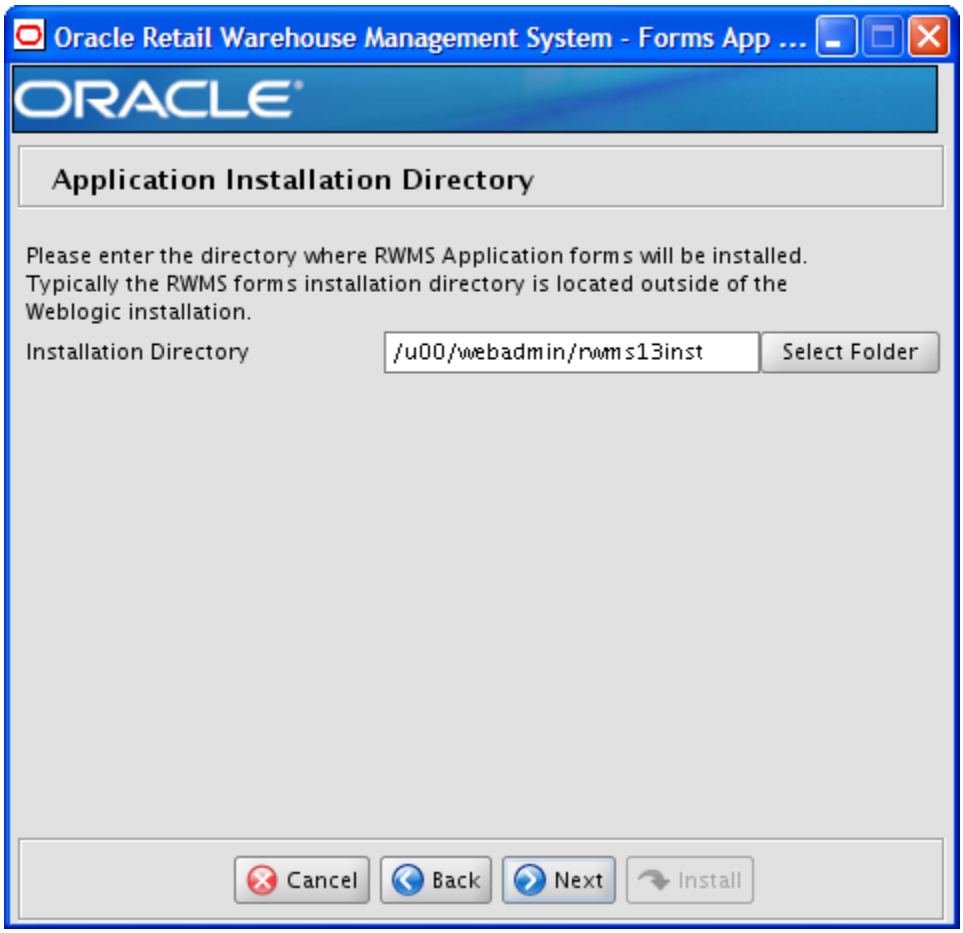
Field Title	Oracle Wallet password
Field Description	This is the password for the wallet that will store the database credentials that were supplied in the Data Source Details screen.

Screen: Installation Name



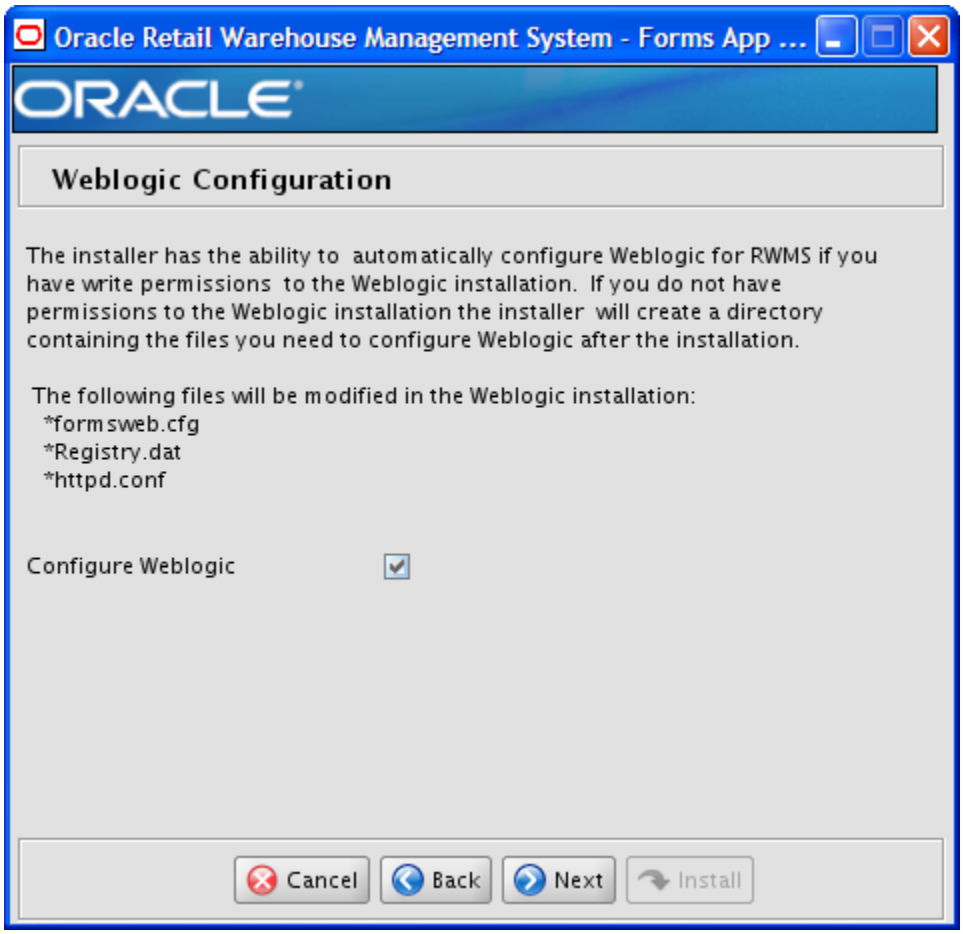
Field Title	Installation Name
Field Description	This value is used in conjunction the Oracle Configuration Manager (OCM). It gives the installation a unique name so the OCM can identify different installations of RWMS in the same WebLogic instance.
Example	rwms13inst

Screen: Application Installation Directory



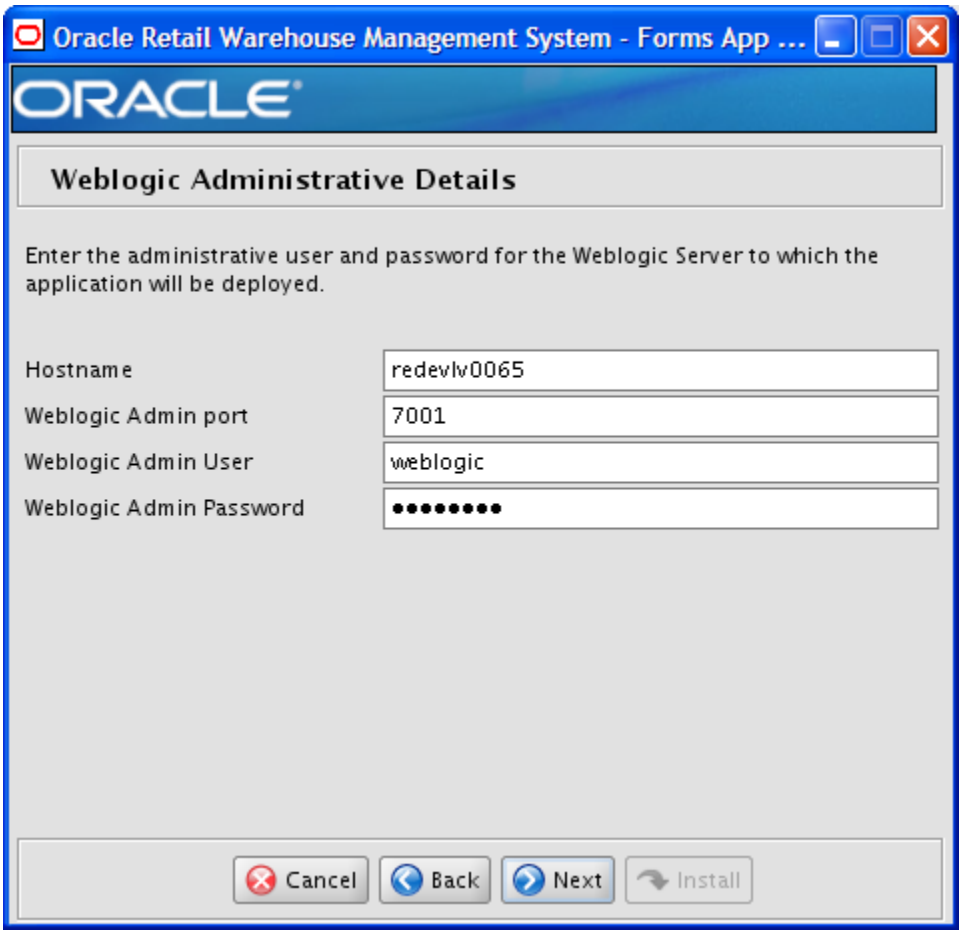
Field Title	Application Installation Directory
Field Description	The location where the RWMS Application (toolset, forms and reports) will be installed. The RWMS $MMHOME path will be a subdirectory of this directory, named base.
Example	/u01/webadmin/rwms13inst

Screen: WebLogic Configuration



Field Title	Configure WebLogic
Field Description	Make the necessary configurations to the WebLogic server to be able to run RWMS forms. If you choose No, these configurations should be done manually. **Note:** If you rerun the installer, and choose to check the box in the installer screens, Configure WebLogic, you may need to clean up duplicate entries in the WebLogic formsweb.cfg file.

Screen: WebLogic Administrative Details



Field Title	Hostname
Field Description	Hostname of the application server
Example	redevlv0065

Field Title	WebLogic Admin port
Field Description	Port number of the WebLogic AdminServer
Example	7001

Field Title	WebLogic Admin User
Field Description	Username of the admin user for WebLogic instance to which the RWMS Webhelp application is being 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.

C

Appendix: Common Installation Errors

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

GUI Mode Crashes when Installing on AIX 7.1

Symptom

There is a known issue with the installer on AIX7.1. The installer in GUI mode will crash when it communicates with Oracle database, and produces two binary dump files (core.<timestamp>.dmp, Snap.<timestamp>.trc) and a javacore text file (javacore.<timestamp>.txt).

Solution

As a workaround, please run the installer in text mode (ksh install.sh text) or silent mode (ksh install.sh silent).

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.

Unreadable buttons in the Installer

If you are unable to read the text within the installer buttons, it probably means that your JAVA_HOME is pointed to a pre-1.4.2 JRE or JDK.

Set JAVA_HOME with the appropriate JDK (the same JDK that has been used by WebLogic Server).

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.

Warning: Could not find X Input Context

Symptom

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

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

Solution

This message is harmless and can be ignored.

Unresponsive drop-downs

Symptom

In GUI mode, when you click on the drop-down list, the list does not appear, and the following message appears in the console window:

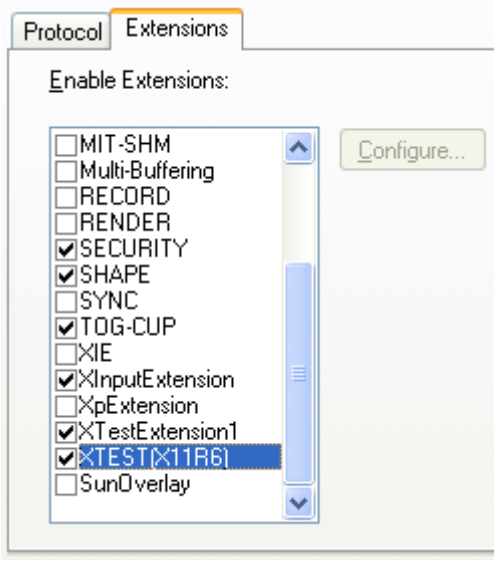
```
XTEST extension not installed on this X server: Error 0
```

Solution

To run the installer in GUI mode, you must have the XTEST extension enabled in your X server.

Enabling XTEST in Exceed:

1. Open Xconfig to edit Exceed configuration.
2. Go to the X Server Protocol settings.
3. Click the Extensions tab.
4. Make sure that the XTEST extension is selected:



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

Could not 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.

FRM-30064: Unable to parse statement select while compiling fm_ituda.fmb

Symptom

When running the application installer you get the following error:

```
FRM-30064: Unable to parse statement select vu.uda_desc,  vu.uda_id  from v_uda vu
where get_primary_lang = get_user_lang  and vu.display_type = 'LV' union all
select nvl(t.translated_value, vu.uda_desc),  vu.uda_id  from tl_shadow t,  v_uda
vu  where get_primary_lang != get_user_lang and upper(vu.uda_desc) = t.key(+) and
get_user_lang = t.lang(+) and vu.display_type = 'LV' order by 1.
ORA-28112: failed to execute policy function
Record Group RG_UDA_LOV
Form: FM_ITUDALST

FRM-30085: Unable to adjust form for output.

Form not created
```

Solution

Disable the database filter policies by running drop_filter_policy.sql, run the application installer again and then run add_filter_policy.sql. Both files can be located with the database installer.

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 servers 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 server 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.

Forms Installer fails on HP-UX

Symptom

Errors occur during Forms installer screens when run on HP-UX. When you click **Next** on the installer screen, Data Source Details, you get this error message on the screen: "no ocijdbc11 in java.library.path." The message prevents you from moving to the next screen.

Solution

This error message can be ignored. Verify that the data source details you entered are correct, and uncheck the box labeled **Test Data Source?** The installer screens will not attempt to validate the data source when you click **Next**. But the installer will attempt to validate once again when installation starts, and the installer will fail if the credentials are incorrect.

Deployed BI Publisher application fails to start up

Symptom

After deploying BI Publisher into the WebLogic, the application fails to start up with a "missing CipherException.class" exception.

Solution

Add CLASSPATH to point to the osdt_core3.jar which resides in $WLS_HOME/oracle_common/oui/jlib/lib.

Go to the WebLogic Adminconsole > BIPublisher managed server that is used for deploying BI Publisher. Click the Server Start tab. In the Class Path box, add the following (values are examples):
/u00/webadmin/product/WLS/oracle_common/oui/jlib/lib

RWMS Forms recompile could fail although a successful message appears

Symptom

RWMS forms could fail during installation, but a successful message is displayed when the installation is finished. There may be some errors in the log file.

Solution

Verify the logs after RWMS forms compilation has been completed for failure messages.

RWMS Forms apply_item_class_s.fmb or fwd_pick_loc_editor_s.fmb fail to compile

Symptom

RWMS forms apply_item_class_s.fmb or fwd_pick_loc_editor_s.fmb fail during installation, with the message"Compilation errors have occurred. Form not created."

Solution

Recompile the plsql packages ADD_BY_POLIST and FPR_PO_ASSIGN_S in the RWMS schema and recompile the forms.

Example: ALTER PACKAGE FPR_PO_ASSIGN_S COMPILE PACKAGE;

GUI screens fail to open when running Installer

Symptom

When running the installer in GUI mode, the screens fail to open and the installer ends, returning to the console without an error message. The ant.install.log file contains this error:

```
Fatal exception: Width (0) and height (0) cannot be <= 0
java.lang.IllegalArgumentException: Width (0) and height (0) cannot be <= 0
```

Solution

This error is encountered when Antinstaller is used in GUI mode with certain X Servers. To workaround this issue, copy ant.install.properties.sample to ant.install.properties and rerun the installer.

D

Appendix: URL Reference

This appendix is a reference for URL information.

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

For 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

For example, jdbc:oracle:thin:@myhost:1521:mysid1:mysid

LDAP Server URL

Used by the Java application to connect to the LDAP directory.

Syntax: ldap://<host>:<port>

<host>: hostname of the directory server

<port>: LDAP server port

For example, ldap://myhost:389t:389

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 WebLogic Server documentation for details.

E

Appendix: Oracle Single Sign-On Resource Access Descriptors

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

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

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

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

RADs may be created and through LDIF scripts as well. For information, see the My Oracle Support document, "How to Create and Copy SSO User Resources (RADs) (ID 244526.1).

F

Appendix: Oracle 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 currently provides two different implementations: Oracle Single Sign-On (OSSO), and Oracle Access Manager (provides more comprehensive user access capabilities).

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

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

Most SSO implementations are performed in an application's infrastructure and not in the application logic itself. Applications that leverage infrastructure managed authentication (such as 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 Oracle Single Sign-On?

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

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle HTTP Server 11g Release 1 as a front end to the Oracle WebLogic Server. The Oracle HTTP Server is included in the Oracle Web Tier Utilities 11g Release 1 (11.1.1).
- An Oracle Single Sign-On Plug-in, used to authenticate the user and create the OSSO session cookie. This is available in the Oracle Fusion Middleware 11g Web Tier Utilities (11.1.1.6) package. For Oracle Forms applications like RMS and RWMS, HTTP server will be used.
- The Delegated Administration Services (DAS) application in OID10g and 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 OSSO system and registering HTTP servers.

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

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

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

Oracle Single Sign-on Terms and Definitions

The following terms 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 OSSO environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic OSSO protection typically display a Login link to provide user authentication and gain greater access to the application's resources.

Identity Management Infrastructure for 10g, Oracle Identity Management (OIM) and Oracle Access Manager (OAM) Oracle Access Manager (OAM) for 11g

If using OSSO 10g, The Identity Management Infrastructure is the collection of product and services which provide Oracle Single Sign-on functionality. For OSSO 10g, this includes the Oracle Internet Directory, an Oracle HTTP server, and the Oracle Single Sign-On services. The Oracle Application Server deployed with these components is typically referred as the Infrastructure instance.

If using SSO with OAM11g, Oracle Identity Management (OIM) 11g includes Oracle Internet Directory and ODSM. Oracle Access Manager (OAM) 11g should be used for SSO using osso agent. Oracle Forms 11g contains Oracle HTTP server and other Retail Applications will use WebTier11g for HTTP.

MOD_OSSO

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

MOD_WEBLOGIC

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

Oracle Internet Directory

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

Partner Application

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

All partner applications must be registered with the Oracle Single Sign-On server if using OSSO10g and all partner applications must be registered with Oracle Access Manager (OAM) 11g if using OAM11g for SSO implementation. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Realm

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

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

Statically Protected URLs

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

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

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

What Single Sign-On is not

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

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

How Oracle Single Sign-On Works

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

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

Statically Protected URLs

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

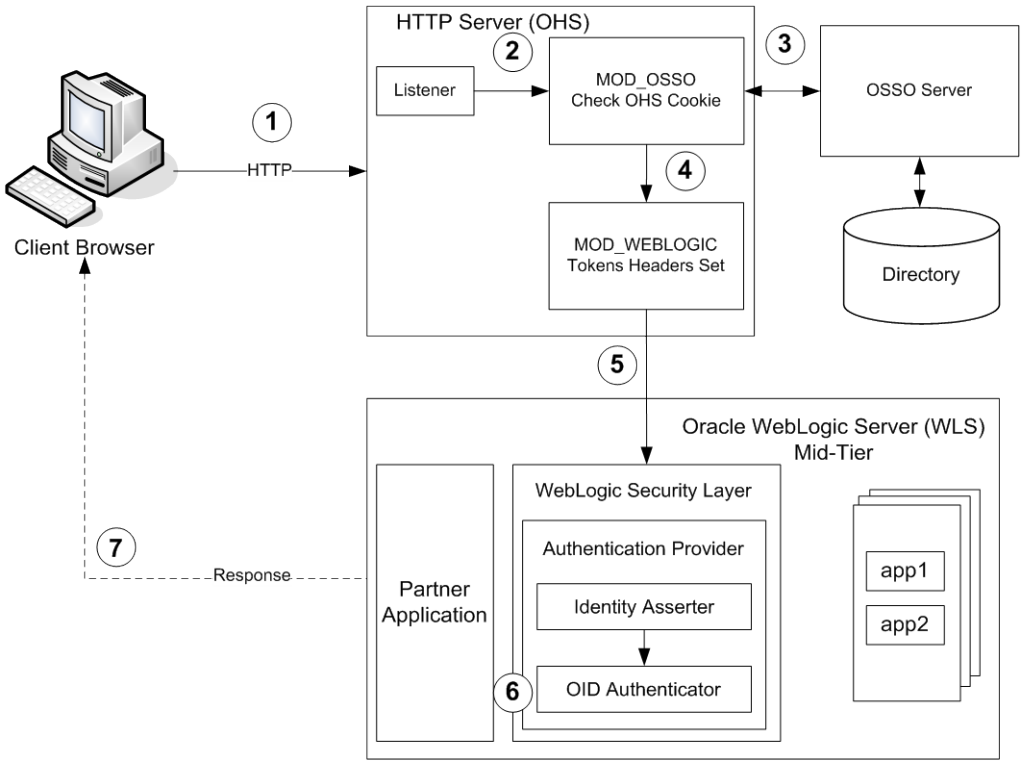
1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS).
2. The Oracle HTTP Server processses the request and routes it to the mod_oss module.
3. This module determines whether the user is already authenticated. If the authentication is required, it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

Dynamically Protected URLs

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

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

Single Sign-on Topology



Installation Overview

Installing Oracle Single Sign-On 10g requires installation of the following:

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

   The Infrastructure OAS includes the Delegated Administration Services (DAS) application as well as the OSSO servlet. The DAS application can be used for user and realm management within OID.
2. 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 Infrastructure OAS installed in step 1. For additional information on SSO 10g installation, see the Creating a High-Availability Environment Whitepaper (My Oracle Support Doc ID: 1311392.1).
3. Additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities.

Installing Oracle Single Sign-On 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. For additional information on SSO 11g installation, see the Oracle Access Manager and Single Sign-On Whitepaper (My Oracle Support Doc ID 1492047.1).

Infrastructure Installation and Configuration

The Infrastructure installation for OSSO and Oracle Access Manager (OAM) is dependent on the environment and requirements for its use. Deploying an Infrastructure OAS or 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 Application Server Installation Guide and the Oracle Internet Directory Installation Guide (if using OSSO 10g) for more details and Oracle Identity Management Installation Guide11g (if using OAM11).*

OID User Data

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

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

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

OID with Multiple Realms

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

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

User Management

User Management consists of displaying, creating, updating or removing user information. There are two basic methods of performing user management: LDIF scripts and the Delegate Administration Services (DAS) application available for OID10g or Oracle Directory Services Manager (ODSM) available for OID11g.

OID DAS

The DAS application is a Web-based application used in OID10g is designed for both administrators and users. A user may update their password, change their telephone number of record, or modify other user information. Users may search for other users based on partial strings of the user's name or ID. An administrator may create new users, unlock passwords, or delete users.

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

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

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

G

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 os the setup you will be able to enter a connect string, such as sqlplus /@<db alias from tnsname.ora>.

For example:

sqlplus /@dvols29_wms01user

Set up a Wallet

To set up a wallet, complete the following steps.

1. Create a new directory called wallet under the folder structure where you run RWMS batch.
   - cd /projects/rwms13.2/dev/forms
   - mkdir wallet
   - 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/rwms13.2/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/11.2.0.1/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 to not 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/rwms13.2/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/rwms13.2/dev/forms

   $ sqlplus /@dvols29_wms01user

   SQL*Plus: Release 11.2.0.1.0 - 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 11.2.0.1.0 - 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

The following are more 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/rwms13.2/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 with exporting UP to show schema/password@db. Using wallet, we are able to keep the password invisible.

Set 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/rwms13.2/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 this command:

   ./save_credential.sh -u <USERNAME> -a <ALIASNAME> -p <PARTITION NAME> -l <WALLET PATH>

   Example: ./save_credential.sh -u wms01user -a RWMS132_USER -p rwms132 -l /projects/rwms13.2/extras/javawallet

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

   For example, password is retek@sid.

   ./save_credential.sh -u Administator-a BI_ALIAS -p rwms132 -l /projects/rwms13.2/extras/javawallet

   Password could be Administrator.
6. Re-run 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 the following example.

   rwms13inst.env

```
--------------
TNS_ADMIN=/projects/rwms13.2/dev/forms/wallet
RWMS_WALLET_PATH=/projects/rwms13.2/extras/javawallet
RWMS_WALLET_PARTITION=rwms132
RWMS_WALLET_LOGON=TRUE
RWMS_BI_PWD=BI_ALIAS
RWMS_DB_CONNECT= RWMS132_USER

CLASSPATH=$CLASSPATH:/projects/rwms13.2/extras/wmsSecurity.jar
```

H

Appendix: Setting Up Password Stores with Oracle Wallet

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

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

About Password Stores and Oracle Wallet

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

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

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

```
sqlplus /@db_username
```

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

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

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

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

Setting Up Password Stores for Database User Accounts

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

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

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

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

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

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

For RMS, RPM Plsql Batch, RETL DB, RWMS batch, and ARI, do the following.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

   $ sqlplus /@dvols29_rms01user

   SQL*Plus: Release 11

   Connected to:
   Oracle Database 11g

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

   Running batch programs or shell scripts would be similar:

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

   Set the UP unix variable to help with some compiles :

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

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

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

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

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

- List the wallet credential entries

  ```
  mkstore –wrl . –list
  ```

  This command returns values such as the following.

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

- View the details of a wallet entry

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

  Returns the value of the entry:

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

  Returns value of the entry:

  ```
  rms01user
  ```

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

  Returns value of the entry:

  ```
  passwd
  ```

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

For Java application, consider the following:

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.
- Passwords are not used to update entries in Java wallets. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application that has been installed, the wallet is located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/config Example: mspdv351:[103x_WLS] /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config
- Application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.
- Scripts are located in <WEBLOGIC_DOMAIN_HOME>/retail/<appname>/retail-public-security-api/bin for administering wallet entries.

  Example:

  mspdv351:[103x_WLS] /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/retail-public-security-api/bin

- In this directory is a script to help you update each alias entry without having to remember the wallet details. For example, if you set the RPM database alias to rms01user, you will find a script called update-RMS01USER.sh.

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

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

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

  Example:

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

update-<ALIAS>.sh

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

Usage:

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

Example:

```
mspdev71:[103xWLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/java_domain/retail/rpm132test/retail-public-security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass in.
mspdev71:[103xWLS]
/u00/webadmin/product/10.3.x/WLS/user_projects/domains/java_domain/retail/rpm132test/retail-public-security-api/bin>
```

dump_credentials.sh

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

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

Example:

dump_credentials.sh /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

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

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

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

save_credential.sh

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

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

Example:

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

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

Enter password:
Verify password:
```

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

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

Usage

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

How Does the Wallet Relate to the Application?

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

Reim.properties code sample:

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

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

How Does the Wallet Relate to Java Batch Program Use?

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

Setting up RETL Wallets

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

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

To set up RETL wallets, perform the following steps:

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

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

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

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

   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     ```
     export RETL_WALLET_ALIAS="retl_java_rms01user"
     ```
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     ```
     export ORACLE_WALLET_ALIAS="dvols29_rms01user"
     ```
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     ```
     export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"
     ```

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

Quick Guide for Retail Wallets

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (MMHOME)>/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile, execution	Installer	n/a	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile	Installer	n/a	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	n/a	<Db_Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rwms schema owner>	Compile forms, execute batch	Installer	n/a	Alias hard-coded by installer
RPM app	DB	<RPM batch install dir>/.wallet	n/a	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	
RWMS auto-login	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	<RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms13inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	n/a	Alias hard-coded by installer
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip13	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip13	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
			aip13	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm13	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rpm13	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			rpm13	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbatch-alias	

Retail App	Wallet Type	Wallet Location	Wallet Partition	Alias Name	User Name	Use	Create By	Alias Example	Notes
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/config							Each alias must be unique
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
			<installed app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
RSL app	JAVA	<RSL INSTALL DIR>/rsl-rms/security/config							Each alias must be unique
			rsl-rsm	<rsl weblogic user alias>	<rsl weblogic user name>	App use	Installer	weblogic-alias	
			rsl-rsm	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
SIM app	JAVA	<weblogic domain home>/retail/<deployed sim app name>/config							
			rpm	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rms	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			rib-sim	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application

Retail App	Wallet Type	Wallet Location	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	

I

Appendix: Optional Security

The following scripts could be used to lock down permissions on the RWMS owning schema. Note that the grant script must be run before applying each hot fix or patch to the schema.

(Optional) Grant Necessary Install Permissions

If you ran the optional revoke_rwms_admin_privs.sql script to secure the permissions for the RWMS schema, you must grant additional permissions before running the database installation.

To grant necessary install permissions to the RWMS schema, run the following script as the database sys user.

DB_PATCH_DIR/rwms/dbschemapatch/util/grant_rwms_admin_privs.sql

> **Note:** After completing installation, run the revoke script to restore the optional secure permissions on the RWMS schema.

(Optional) Revoke Install Permissions

After completing database installation, you can secure the RWMS schema by granting only the minimum required permissions. To do this, run the following script as the database sys user.

DB_PATCH_DIR/rwms/dbschemapatch/util/revoke_rwms_admin_privs.sql

Recompile Invalid Objects

After running the revoke script, recompile all invalid database objects.

J

Appendix: Web Browser Configuration

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

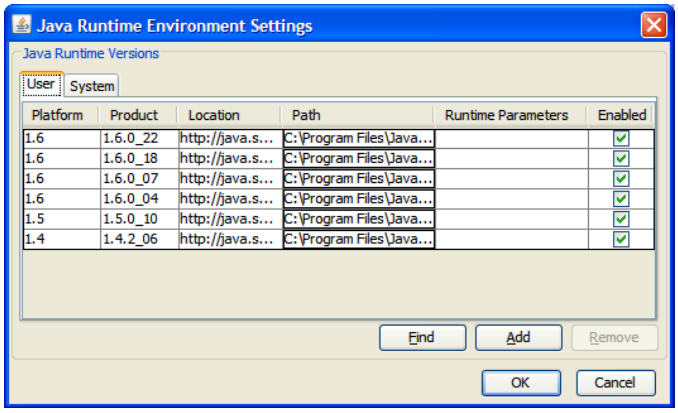
> **Note:** The Oracle's Java Runtime Environment (JRE) is required to support Applets within a sandboxed security environment in the Web browser. The security architecture of the JRE has changed from JRE1.6.0_18 and requires additional configuration in Windows.

Microsoft Internet Explorer Version 8.0 or 9.0

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

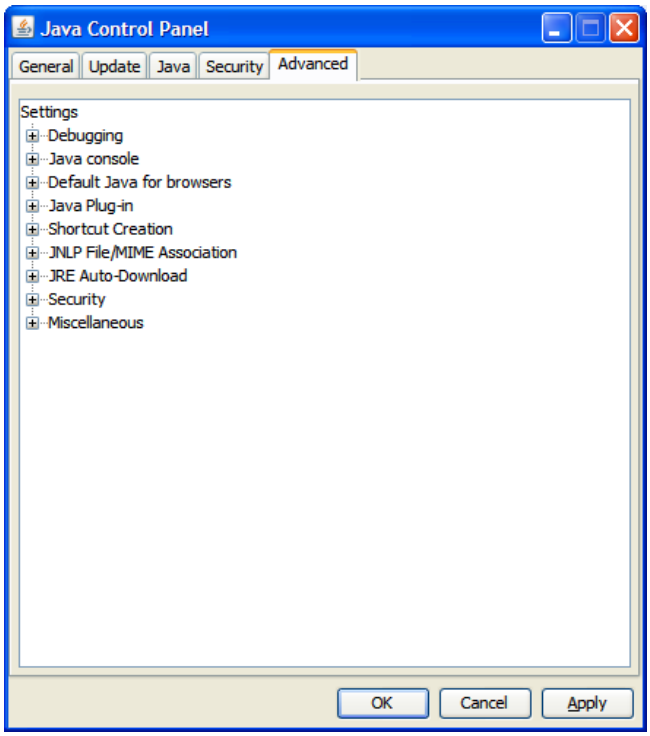
To configure the latest JRE for Internet Explorer 8, do the following.

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



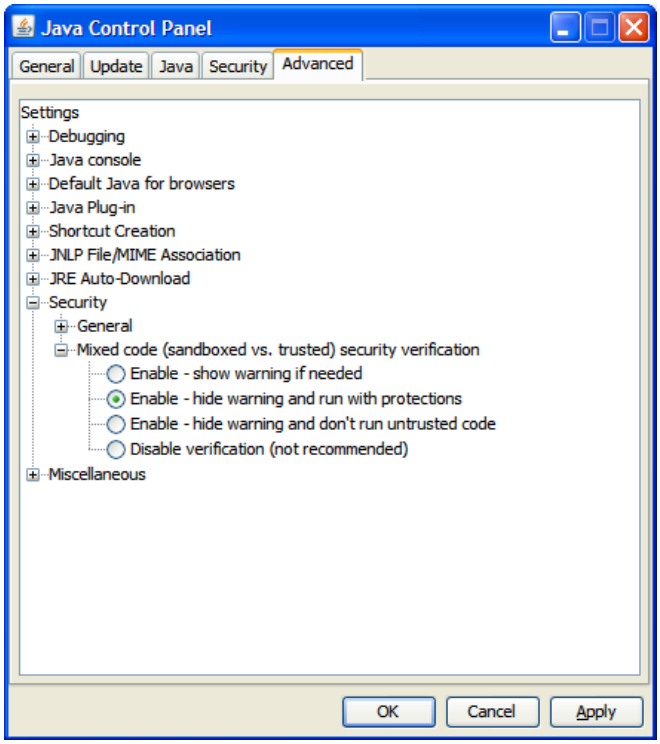
Java Runtime Environment Settings Window

1. Click **OK** and return back to the Java Control Panel.
2. Click the **Advanced** tab.



Options in the Advanced Tab

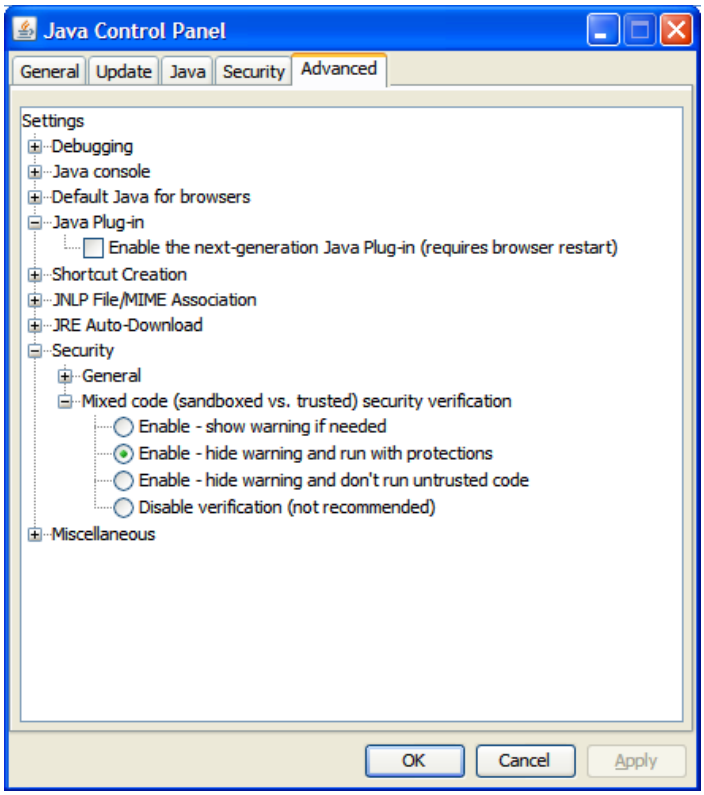
3. Click the Plus (+) icon and expand **Security**, and then expand **Mixed code**.



Mixed Code Settings in Java Control Panel

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

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



Java Plug-in Option in Java Control Panel

Restart your Web browser to run the RWMS Forms displays.

Mozilla Firefox Version 3.6.x, 2.3 or 10.0

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).

K

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) or Back Office with Labels and Tags (ORLAT)
12. Oracle Retail Store Inventory Management (SIM)

    > **Note:** During installation of SIM, you are asked for the RIB provider URL. 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 Replenishment Optimization (RO)
17. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)

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