Oracle® Retail Merchandising System

Installation Guide
Release 13.2
E22174-02

March 2011

ORACLE

Oracle® Retail Merchandising System Installation Guide, Release 13.2

Copyright © 2011, Oracle. All rights reserved.

Primary Author: Nathan Young

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

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

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

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

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

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

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

Contents

Send Us Your Comments ..... **xi**

Preface ..... **xiii**

Audience xiii
Related Documents xiii
Customer Support xiv
Review Patch Documentation xiv
Oracle Retail Documentation on the Oracle Technology Network xiv
Conventions xiv

1 Preinstallation Tasks **1**

Check for the Current Version of the Installation Guide 1
Implementation Capacity Planning 1
Hot Fixes Available for RMS 13.2 1
A Note to Retailers Using the Croatian Language 2
Check Supported Database Server Requirements 2
Check Supported Application Server Requirements 4
Verify Single Sign-On 4
Check Supported Web Browser and Client Requirements 5
Supported Oracle Retail Products 5
Supported Oracle Retail Integration Technologies 5
Supported Oracle Applications 6
UNIX User Account Privileges to Install the Software 6
Verify RMS and SIM Inventory Adjustment Reason Codes 6

2 RAC and Clustering **7**

Part I: Full Installation **9**

1 Database Installation Tasks—Full **11**

RMS 13.2 Full Release 11
RMS Database Schema Distribution – Oracle Retail Applications Included 11
Create Staging Directory for RMS Database Schema Files 11
Establish Database Partitioning Strategy 11
Step 1: Modify partition_attributes.cfg 13
Step 2: Modify Data Definition Files 13
Step 3: Generate DDL for Tables – Run partition.ksh 14
Create the RMS Database 15
Create the Database Instance Using a Template 15
Create the Tablespaces 34
Create the Schema Owner for RMS 35
Create the Database User for Allocation 35
Review RIB CLOB Settings 35
Run the RMS Database Schema Installer 36

Values to Remember for the Batch and Application Installers....37
Resolving Errors Encountered During Database Schema Installation....37
Set Up Additional RMS Users....37
2 Batch Installation Tasks—Full....39
Create Staging Directory for RMS Batch Files....39
Run the RMS Batch Installer....39
Resolving Errors Encountered During Batch Installation....40
RETL....40
Data Conversion Scripts....41
3 Application Server Installation Tasks—Full....43
Prepare Application Server for RMS....43
Create RMS Help Managed Server....44
Install Node Manager....48
Create Staging Directory for RMS Application Server Files....54
Run the RMS Application Installer....54
Resolving Errors Encountered During Application Installation....56
Clustered Installations – Post-Installation Steps....57
Oracle Configuration Manager....57
RMS Reports Copied by the Application Installer....57
Test the RMS Application....57
4 RMS Reports—Full....59
Prerequisites....59
Path References Used in this Section....59
BI Server Component Installation Tasks....60
Installation Process Overview....60
Extracting the BI Server Web Archive from the Source Media....60
Creating an Exploded Directory for the Installation....61
Configuring the BI Publisher Repository and Installing Fonts....62
Install Managed Server in WebLogic....63
Install Node Manager....68
Start the Managed Servers....72
Additional Set Up Steps Before Deploying the BI Application....72
Deploying the BI Application in WebLogic....74
Installing the RMS BI Publisher Templates....80
Configuring the RMS JDBC connection....81
Configuring the BIPublisher Scheduler....82
Part II: Upgrade Installation....83
1 RMS Database Installation—Upgrade....85
Preparing to upgrade the RMS Database....85
Option 1: Upgrade RMS Database using the Installer....85
Create Staging Directory for RMS Database Schema Files....86
Run the RMS Database Schema Installer....86

Resolving Errors Encountered During Database Schema Installation 87
Option 2: Upgrade RMS Database using Controller Scripts 87
Create Staging Directory for RMS Database Schema Files 88
Run the RMS Database Controller Scripts 88
2 Batch Installation Tasks—Upgrade 91
Option 1: Use Batch Installer to Upgrade 91
Run Batch Installer 91
Option 2: Compile RMS Batch Directly 91
Create Staging Directory for RMS Batch Upgrade Files 91
Set Environment Variables 92
Compile Batch Libraries 92
Compile Batch Source Code 93
3 Application Server Installation Tasks—Upgrade 95
Option 1: Use Application Installer to Upgrade 95
Run Application Installer 95
Test the RMS Application 95
Option 2: Compile RMS Toolset and Forms Directly 96
Create Staging Directory for RMS Application Upgrade Files 96
Set Environment Variables 96
Install RMS Toolset 97
Install RMS Forms 99
Install Helpfile 100
4 RMS Reports Installation—Upgrade 101
5 Data Migration 103
Create Staging Directory for RMS Data Migration Files 103
Configure RMS Data Migration Tool 103
Run the RMS Data Migration Tool 105
Configure ReIM Data Migration Tool 106
Run the ReIM Data Migration Tool 107
Configure Allocation Data Migration Tool 107
Run the Allocation Data Migration Tool 108
6 Web Services Installation 109
Extract Web Services Files 109
Create a Managed Server 109
Create a Datasource 109
Deploy RMS Service EAR File 109
Configure Web Service Security 111
Attach Policy File to the Web Service 112
Create Roles and Users who can Access the Web services 113
A Appendix: Oracle 11g Database Parameter File 115
B Appendix: Configure Listener for External Procedures 117

C Appendix: Tablespace Creation Scripts 119
D Appendix: RMS User Creation Script 121
E Appendix: RMS RETL Instructions 123
Configuration 123
RETL 123
RETL User and Permissions 123
F Appendix: Oracle Trade Management System Expectations 125
Installation Scripts (elc_comp_post_htsupld.sql) 125
HTS Upload / Mass Update 127
Calculation of Merchandise Processing Fee 128
Unit of Measure Conversions 128
Customs Entry Ref. Status 128
Customs Entry Totals 129
G Appendix: RMS Database Schema Installer Screens 131
H Appendix: RMS Batch Installer Screens 159
I Appendix: RMS Application Installer Screens 165
J Appendix: Installer Silent Mode 177
Repeating an Installation Attempt 177
K Appendix: URL Reference 179
JDBC URL for a Database 179
LDAP Derver URL 179
JNDI Provider URL for an Application 179
L Appendix: Common Installation Errors 181
Database Installer Hangs on Startup 181
Unreadable Buttons in the Installer 181
Warning: Could Not Create System Preferences Directory 181
Warning: Could Not Find X Input Context 182
Unresponsive Country and Currency Drop-Downs 182
Could Not Execl Robot Child Process: Permission Denied 183
ConcurrentModificationException in Installer GUI 183
FRM-30064: Unable to Parse Statement Select While Compiling fm_ituda.fmb 183
ORA-04031 (Unable to Allocate Memory) Error During Database Schema Installation 184
X Error of Failed Request: BadWindow (Invalid Window Parameter) 184
RIB Errors 184
Error Connecting to Database URL 185
Multi-Threaded OCI Client Dumps Core after Reconnecting To Database 185
Forms Installer Fails on HP-UX 186
Manually Running partition.ksh Script Fails 186

M Appendix: Manual Forms Compilation 187
Set Environment Variables 187
Compile RMS Toolset 188
Compile RMS Forms 190
N Appendix: Application Deployment Method 191
O Appendix: Manual Batch Compilation 193
Set Environment Variables 193
Configure Make File 194
Create Batch Libraries in Database 194
Revalidate RMS Database Objects 195
Compile Batch Libraries 195
Compile Batch Source Code 195
P Appendix: Oracle Single Sign-On 197
What Do I Need for Oracle Single Sign-On? 197
Can Oracle Single Sign-On Work with Other SSO Implementations? 197
Oracle Single Sign-on Terms and Definitions 198
What Single Sign-On is not 199
How Oracle Single Sign-On Works 199
Installation Overview 201
User Management 202
Q Appendix: Single Sign-On Resource Access Descriptors 205
R Appendix: AIX Shared Library Bug Fix 207
S Appendix: Inserting New Languages 209
Insert Secondary Language Data 209
Insert Primary Language Data 210
T Appendix: Setting Up Password Stores with Oracle Wallet 211
About Password Stores and Oracle Wallet 211
Setting Up Password Stores for Database User Accounts 212
Setting Up Wallets for Database User Accounts 213
For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI 213
For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL) 215
How does the Wallet Tie Back to the Application? 217
How does the Wallet Tie Back to Java pgm batch Use (such as REIM batch)? 217
Setting up RETL Wallets for 13.2 217
Quick Guide for Retail Wallets 219
U Appendix: Installation Order 225
Enterprise Installation Order 225

Send Us Your Comments

Oracle Retail Merchandising System Installation Guide, Release 13.2

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

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

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

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

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

You can find more information about this product in these resources:

- *Oracle Retail Merchandising System Release Notes*
- *Oracle Retail Merchandising System User Guide* and Online Help
- *Oracle Retail Sales Audit User Guide* and Online Help
- *Oracle Retail Trade Management User Guide* and Online Help
- *Oracle Retail Merchandising System Reports User Guide*
- *Oracle Retail Merchandising System Operations Guide*
- *Oracle Retail Merchandising System Data Model*
- *Oracle Retail Merchandising Batch Schedule*
- *Oracle Retail Merchandising Data Conversion Operations Guide*
- *Oracle Retail Merchandising Implementation Guide*
- Oracle Retail Merchandising Licensing Information
- *Oracle Retail Fiscal Management/RMS Brazil Localization Installation Guide*
- *Oracle Retail Fiscal Management/RMS Brazil Localization Implementation Guide*
- *Oracle Retail Fiscal Management User Guide* and Online Help
- *RMS/ReSA Brazil Localization User Guide* and Online Help
- *Oracle Retail Merchandising System Custom Flex Attribute Solution Implementation Guide*
- *Oracle Retail POS Suite/Merchandising Operations Management Implementation Guide*
- Oracle Retail Integration Bus documentation
- Oracle Retail Service Layer documentation
- Oracle Retail Extract, Transform, and Load documentation

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.1) or a later patch release (for example, 13.1.2). 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 tasks 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.

Implementation Capacity Planning

There is significant complexity involved in the deployment of Oracle Retail applications, and capacity planning is site specific. Oracle Retail strongly suggests that before installation or implementation you engage your integrator (such as the Oracle Retail Consulting team) and hardware vendor to request a disk sizing and capacity planning effort.

Sizing estimates are based on a number of factors, including the following:

- Workload and peak concurrent users and batch transactions
- Hardware configuration and parameters
- Data sparcity
- Application features utilized
- Length of time history is retained

Additional considerations during this process include your high availability needs as well as your backup and recovery methods.

Hot Fixes Available for RMS 13.2

Three hot fixes are available at My Oracle Support to address some issues discovered in RMS 13.2. You must download all of these hot fixes, as applicable for your installation, before you begin to install RMS 13.2:

- Hot fix 11071540

  This hot fix contains miscellaneous changes related to order receiving and the item-supplier data conversion load script. This fix should be applied to all RMS 13.2 installations.

- Hot fix 11071552

  This hot fix contains miscellaneous changes related to pack item upgrades and the entity Clasificación Nacional de Actividades Económicas (CNAE) codes data conversion load script. This fix applies only if you are installing RMS 13.2 with Brazil localization.
- Hot fix 11849744

  This hot fix contains miscellaneous changes related to the RMS 13.2 database and application installers, as well as RMS 13.2 demo data and currency tables. This fix should be applied to all RMS 13.2 installations.

Access My Oracle Support at the following URL:

https://support.oracle.com

Applying the Hot Fixes

See the defect reports and README files included with each hot fix for more information about the content of the fixes and instructions for applying them.

A Note to Retailers Using the Croatian Language

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

Check Supported Database Server Requirements

General requirements for a database server running RMS include the following.

Supported on	Versions Supported
Database Server OS	OS certified with Oracle Database 11gR2 Enterprise Edition. Options are: ▪ Oracle Linux 5 Update 3 (OL5.3) for x86-64 (Actual hardware or Oracle virtual machine). ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) for x86-64 (Actual hardware or Oracle virtual machine). ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or logical domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)

Supported on	Versions Supported
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.1) with the following oneoff patches: ▪ 9582272: ORA-600[kkdlReadOnDiskDefVal: error] occurs when ALTER TRIGGER is executed. ▪ 9100882: SOA: ORA-600[KGHFRE3] SIGNALLED. ▪ 9010222: APPS ST 11G ORA-00600 [KKSFBC-REPARSE-INFINITE-LOOP] ▪ 9932143: [CTS] 3 FAILURES IN JMS/AQ RUN DIDN'T GET EXPECTED MSG BACK AND REDELIVERED FLAG ▪ 9130054: MASSIVE ORA-2051 SIGNALLED DURING SOA TEST AGAINST AN 11.2.0.1 DATABASE. ▪ 9367425: PROCESS CRASHED WHEN USING 11GR2 JDBC/OCI. ▪ 9495959: HANG WHEN TWO THREADS TRY TO CREATE THE ENV HANDLE AT THE SAME. **Note:** Patches 9367425 and 9495959 must be applied together. Components: ▪ Oracle Database 11g ▪ Oracle Partitioning ▪ Oracle Net Services ▪ Oracle Call Interface (OCI) ▪ Oracle Programmer ▪ Oracle XML Development Kit ▪ Examples CD (Formerly the companion CD) Other components: ▪ Perl compiler 5.0 or later ▪ X-Windows interface ▪ ANSI compliant C-compiler (certified with OS and database version).

Check Supported Application Server Requirements

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

Note: Files required for Oracle Configuration Manager (OCM) are removed after OPatch is used to patch a WebLogic server. This will not cause the product installers to fail, but will cause OCM installation to fail. To work around this issue, back up the content of the $ORACLE_HOME/utils/ccr/lib directory prior to applying a patch using OPatch, and recopy the content back after you apply any patches. ORACLE_HOME is the location where WebLogic has been installed.

Supported on	Versions Supported
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.3). Options are: ▪ Oracle Linux 5 Update 3 (OL5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or Logical Domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.3) with the following one off patches ▪ 6880880 – New Opatch version for Linux 64-bit. ▪ 10065423 - MERGE REQUEST ON TOP OF 11.1.1.3.0 FOR BUGS 9891666 9891675 ▪ 9356983 –FORMS FAIL TO COMPILE WITH FRM-30312 WHEN NLS_LANG INCLUDES UTF8 CHARACTERSET **Note**: These patches are for Linux 64-bit only, for other OSes these patches are not required. Patch 10065423 must be applied before patch 9356983. Components: ▪ Oracle WebLogic Server 11g Release 1 (10.3.3) ▪ Oracle Forms Services 11g Release 1 (11.1.1.3) ▪ Other components: ▪ Oracle BI Publisher 10g (10.1.3.4) ▪ Optional (SSO required) ▪ Oracle Internet Directory 10g (10.1.4) ▪ Oracle WebTier 11g (11.1.1.3)

Verify Single Sign-On

If a Single Sign-On is to be used, verify the Oracle Infrastructure Server 11g version 10.1.4 server has been installed. Verify the Oracle WebTier server hosting Oracle Forms is registered with the Infrastructure Oracle Internet Directory.

Check Supported Web Browser and Client Requirements

General requirements for client running RMS include:

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

Supported Oracle Retail Products

Product	Version
Oracle Retail Active Retail Intelligence (ARI)	13.2
Oracle Retail Price Management (RPM)	13.2
Oracle Retail Allocation	13.2
Oracle Retail Invoice Matching (ReIM)	13.2
Oracle Retail Store Inventory Management (SIM)	13.2.1
Oracle Retail Warehouse Management System (RWMS)	13.2
Oracle Retail Advanced Inventory Planning (AIP)	13.2
Oracle Retail Merchandise Financial Planning (MFP)	13.2.1
Oracle Retail Demand Forecasting (RDF)	13.2.1
Oracle Retail Grade	13.2.1
Oracle Retail Predictive Application Server (RPAS)	13.2.1
Oracle Retail POS Suite	13.3

Supported Oracle Retail Integration Technologies

Integration Technology	Version
Oracle Retail Extract, Transform and Load (RETL)	13.2
Oracle Retail Integration Bus (RIB)	13.2
Oracle Retail Service Layer (RSL)	13.2

Supported Oracle Applications

Requirement	Version
Oracle E-Business Suite (General Ledger and Accounts Payable)	Oracle Application Integration Architecture (AIA) Media Pack 2.5 Oracle E-Business Suite integration is supported using the Oracle Financial Operations Control Integration Pack for Oracle Retail Merchandising Suite and Oracle E-Business Suite Financials. See the *Oracle® Application Integration Architecture 2.5: Installation and Upgrade Guide* for specific version information.
PeopleSoft Enterprise Financials	Oracle Application Integration Architecture (AIA) Media Pack 2.5 PeopleSoft Enterprise Financials integration is supported using the Oracle Retail Merchandising Integration Pack for PeopleSoft Enterprise Financials: Financial Operations Control. See the *Oracle® Application Integration Architecture 2.5: Installation and Upgrade Guide* for specific version information.

UNIX User Account Privileges to Install the Software

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

For example, "oretail."

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

Verify RMS and SIM Inventory Adjustment Reason Codes

SIM and RMS must have the same inventory adjustment reason codes to work properly, with the exception of the Pending Reason Code, which is used for internal purposes only.

2

RAC and Clustering

The Oracle Retail Merchandising has been validated to run in two configurations on Linux:

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

The Oracle Retail products have been validated against an 11.2.0.1 RAC database. When using a RAC database, all JDBC connections should be configured to use THIN connections rather than OCI connections

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

References for Configuration:

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

Part I: Full Installation

Part I of this guide details the steps needed to perform a full installation of RMS. Part I contains the following chapters:

- Chapter 3, Database Installation Tasks – Full
- Chapter 4, Batch Installation Tasks – Full
- Chapter 5, Application Server Installation Tasks – Full
- Chapter 6, RMS Reports – Full

For information about an upgrade installation, see Part II.

1

Database Installation Tasks—Full

RMS 13.2 Full Release

RMS 13.2 is a full baseline installation. The RMS 13.1 software should not be installed if installing from this release.

RMS Database Schema Distribution – Oracle Retail Applications Included

The RMS 13.2 Full release contains a database schema installer package that can be used to install the database objects for the following products: RMS, ReSA, RTM, RPM, ReIM, and Allocation.

> **Note:** The Java application installers for RPM, ReIM, and Allocation are separately downloadable under their respective products. It is only the database schema component of these applications that is included with the RMS release.

Create Staging Directory for RMS Database Schema Files

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

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS database schema installation software. There should be a minimum of 1800 MB disk space available in this location.
3. Copy the rms13dbschema.zip file from the RMS 13.2 release to the staging directory. This is referred to as STAGING_DIR when installing database software.
4. Change directories to STAGING_DIR and extract the rms13dbschema.zip file. This creates an rms/dbschema subdirectory under STAGING_DIR.

Establish Database Partitioning Strategy

Partitioning is mandatory for specific tables. Please review this section in its entirety before proceeding with the installation.

Sample Partitioning

The RMS 13.2 database schema installer runs the partitioning script (partition.ksh) automatically using a sample partitioning strategy if you do not run the partition script yourself. This is acceptable for development or demo installations and allows for a simpler installation. However, the resulting partitioning strategy is NOT suitable for production environments. It is highly recommended that the Production Partitioning section below be followed rather than allowing the installer to implement the sample strategy. The installer can be used to install the RMS database schema regardless of the choice made here.

Production Partitioning

Requirements for mandatory and optional partitioning are defined in the Microsoft Excel spreadsheet located in STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/RMS_partition_definition.xls. Since partitioning strategies are complex, this step should be implemented by an experienced individual who has a thorough understanding of partitioning principles and the data to be partitioned.

Use the Microsoft Excel spreadsheet to determine an appropriate partitioning strategy (STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/RMS_partition_definition.xls). The "Partition Method" column indicates the recommended partitioning option(s) for each table. Refer to the information in this file to modify the DDL for partitioned tables. This can be done by manually changing the file STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/rms_part.tab or by implementing the process defined below. This file will be used later in the installation process.

> **Note:** Refer to Oracle11g Database Concepts Chapter 18 "Very Large Databases (VLDB)" for further details regarding partitioning concepts.

Hash partitions: To calculate the number of hash partitions and sub-partitions, enter values for the three parameters highlighted in yellow at the top of the RMS worksheet. Altering these values updates the "Number of Partitions" column for HASH partitioned/sub-partitioned tables. The values in these columns indicate the number of hash partitions/sub-partitions to create.

Partition Factor: This value is used to adjust the number of hash partitions. It is based on the number of active items per location and transactions per location/day. If the number of items/location and/or transactions/store/day is low, the value of partition factor should be high. This calculates fewer hash partitions. The typical factor value ranges from two to four and in special cases, it can be ten or more.

> **Note:** Changing the items/location and transactions/store/day fields on the worksheet does not automatically impact the factor value. They are used as a point of reference only.

Sub-Partition Factor: This value is used to adjust the number of hash sub-partitions. The partition strategy for historical information determines the value of this number. If the number of range partitions is high, the value of sub-partition factor should be high to control the number of sub-partitions. Typically, this value will be 2.

Locations: The total number of active stores and warehouses.

Range partitions: Determine the purging strategy for all of the tables that are RANGE partitioned. Each partition should have a range of multiple key values. For example, if the strategy were to have data available for one year and to purge it every three months, five partitions would be created. In this case, four 3-month partitions and a "max value" partition to contain all data greater than the defined ranges would result. Refer to the "Comments" column and update the value in the "Number of Partitions" column. The value in this column indicates the number of range partitions to create.

List partitions: The DAILY_ITEM_FORECAST and ITEM_FORECAST must be LIST partitioned. If number of partition keys is relatively static, change the value in the Partition Method column to LIST where allowed. This method ensures that each partition key has a separate partition and that none are empty. The Number of Partitions column is automatically updated with the proper number of locations in the event the partition method is changed. The value in this column indicates the number of list partitions to create.

Step 1: Modify partition_attributes.cfg

Modify STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/partition_attributes.cfg based on the partitioning strategy defined in RMS_partition_definition.xls. Changes to this file should be made only as indicated.

partition_attributes.cfg file: (file is comma-delimited)

Sample Entry:

```
ITEM_LOC_HIST,EOW_DATE,RANGE,item_loc_hist.eow_date.date,64,LOC,HASH,item_loc_hist
.loc.number,64,RETEK_DATA
```

Field 1: `Table Name` - *Do not modify*

Field 2: `Partition Key` - *Do not modify*

Field 3: `Partition Method` - Modify based on value in "Partition Method" column in RMS_partition_definition.xls - Valid values are RANGE, LIST, or HASH (case sensitive)

Field 4: `Partition Data Definition Filename` - Do not modify - This field is ignored if Partition Method is not RANGE or LIST

Field 5: `Partition Hash Count` – Modify based on value in "Hash Partitions Calculated" column in RMS_partition_definition.xls. *This field is ignored if Partition Method is not HASH*

Field 6: `Sub-Partition Key` - *Do not modify*

Field 7: `Sub-Partition Method` - Modify based on value in "Sub-partition Method" column in RMS_partition_definition.xls - Valid values are LIST or HASH (case sensitive)

Field 8: `Sub-Partition Data Definition Filename` - Do not modify - This field is ignored if Sub-Partition Method is not RANGE or LIST

Field 9: `Sub-Partition Hash Count` - Modify based on value in "Hash Sub-partitions Calculated" column in RMS_partition_definition.xls. *This field is ignored if Sub-Partition Method is not HASH*

Field 10: `Tablespace Name` - Optional. Default is RETEK_DATA

Step 2: Modify Data Definition Files

Tables partitioned or sub-partitioned by RANGE or LIST have a corresponding data definition file in the STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/data_def directory and should not be removed or renamed. These files are used to define the data boundaries for each partition. Values must be entered in each file based on the data type of the "Partition Key" column in RMS_partition_definition.xls. Refer to the "Comments" column in this file for additional information. The value in the "Number of Partitions" column indicates the number of entries to place in the data definition file.

The format of a data definition file name is <table name>.<partition key column>.<partition key data type>, for example, item_loc_hist.eow_date.date. When entering data into these files, enter one data partition value per line.

When entering varchar2 values in a data definition file, do not use quotation marks. When defining date values, use the DDMMYYYY format.

sampletable.action_date.date:

```
01012004
01012005
```

sampletable.state.varchar2:

```
Minnesota
Iowa
```

sampletable.location.number:

```
1000
2000
```

When using RANGE partitioning, the data definition files will use the "value less than" concept. For example, in sampletable.action_date.date above, the first partition contains all data less than 01012004. The second partition contains all data greater than or equal to 01012004 and less than 01012005. A third "MAXVALUE" partition is automatically created for all data greater than or equal to 01012005.

When using LIST partitioning, the data definition files use the "value equal to" concept. For example, in sampletable.state.varchar2 above, the first partition will contain all data equal to Minnesota. The second partition will contain all data equal to Iowa.

Step 3: Generate DDL for Tables – Run partition.ksh

```
Change 2 file values in  STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/
partition_attributes.cfg
PREPARTDDL=./rms13.tab
PARTDDL=./rms13_part..tab
Should be valued
PREPARTDDL=./rms.tab
PARTDDL=./rms_part..tab
```

to feed the file names to the RMS installer properly.

Execute STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/partition.ksh at the UNIX command prompt. This script reads configuration information from the partition_attributes.cfg file and generates the partitioned DDL file STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/rms_part.tab. This file is used later during the installation process.

Sample output from partition.ksh:

```
<STAGING_DIR>/rms/dbschema/dbscripts_rms/ddl/part > ./partition.ksh
###############################################################################
# partition.ksh:
# This script will read the partition_attributes.cfg file and any referenced
# data definition files and generate partitioned DDL.
###############################################################################
# The non-partitioned DDL file is ../rms.tab.
# The partitioned DDL file that will be generated is ../rms_part.tab.
###############################################################################
Checking partition_attributes.cfg for errors
Generating Partitioned DDL for DAILY_DATA
Generating Partitioned DDL for DAILY_ITEM_FORECAST
Generating Partitioned DDL for DAILY_SALES_DISCOUNT
…
partition.ksh has generated the DDL for partitioned tables in the ../rms_part.tab
file.
Completed successfully
```

Create the RMS Database

It is assumed that Oracle 11g release 2, with appropriate patches, has already been installed. If not, refer to Check Supported Database Server Requirements in Chapter 1 before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in Create Staging Directory for RMS Database Schema Files in Part I, Chapter 1.

Review the Establish Database Partitioning Strategy section before continuing.

If a database has already been created, it is necessary to review the contents of this section to determine if all database components have been installed and configured properly. Also refer to appendices A, B, C, D, and E in this document.

If a database instance has not been created, create one using one of the methods shown below. As of 13.2, Retail databases are created using database creation templates via DBCA either in interactive mode or in silent mode.

Create the Database Instance Using a Template

Before trying to create databases using a template, the following files must be present in the $ORACLE_HOME/assistants/dbca/templates directory:

- Retail_DB_Template_13.2_OS_Platform_Release.dfb
- Retail_DB_Template_13.2_ OS_Platform _Release.ctl
- Retail_DB_Template_13.2_ OS_Platform _Release_variables.txt
- Retail_DB_Template_13.2_ OS_Platform _Release.dbc

If the files are not there, copy Retail_DB_Template_13.2_ OS_Platform _Release.tar.gz from STAGING_DIR/rms/dbschema/create_db into this directory then unzip and untar to extract the file as follows:

```
gzip –d Retail_DB_Template_13.2_ OS_Platform _Release.tar.gz
tar –xvf  Retail_DB_Template_13.2_ OS_Platform _Release.tar
```

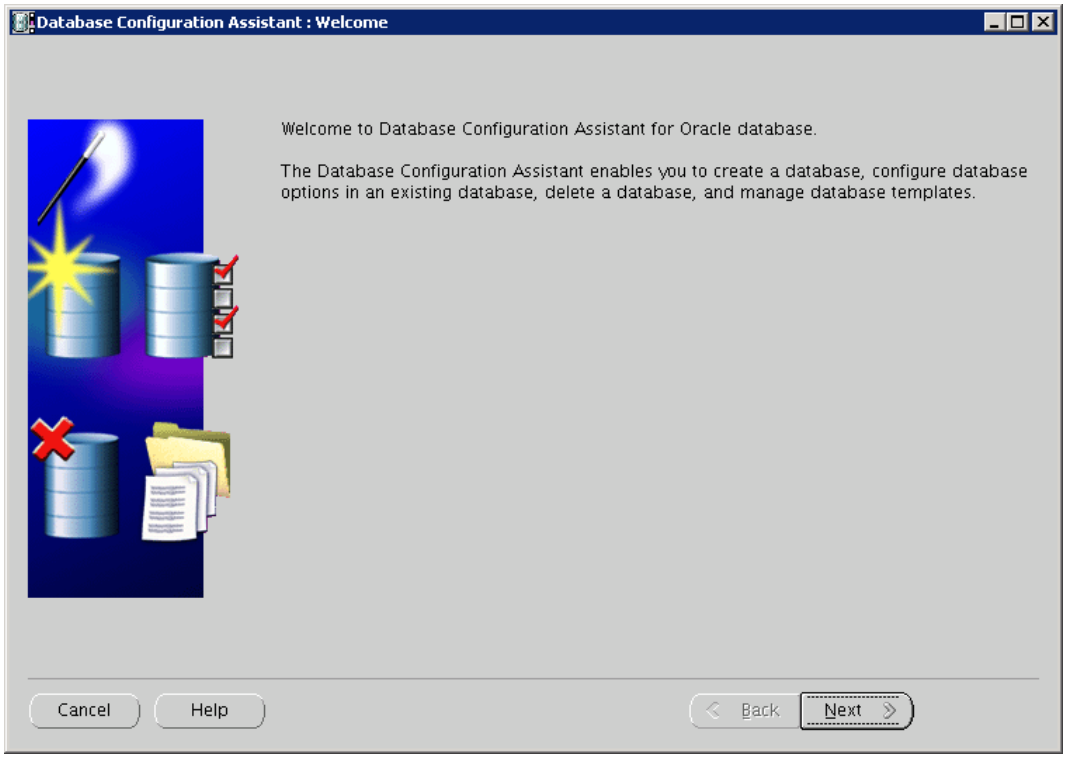
Interactive Instance Creation Mode Using DBCA

1. Login to UNIX as the oracle user; typically the user that owns the Oracle Database software.

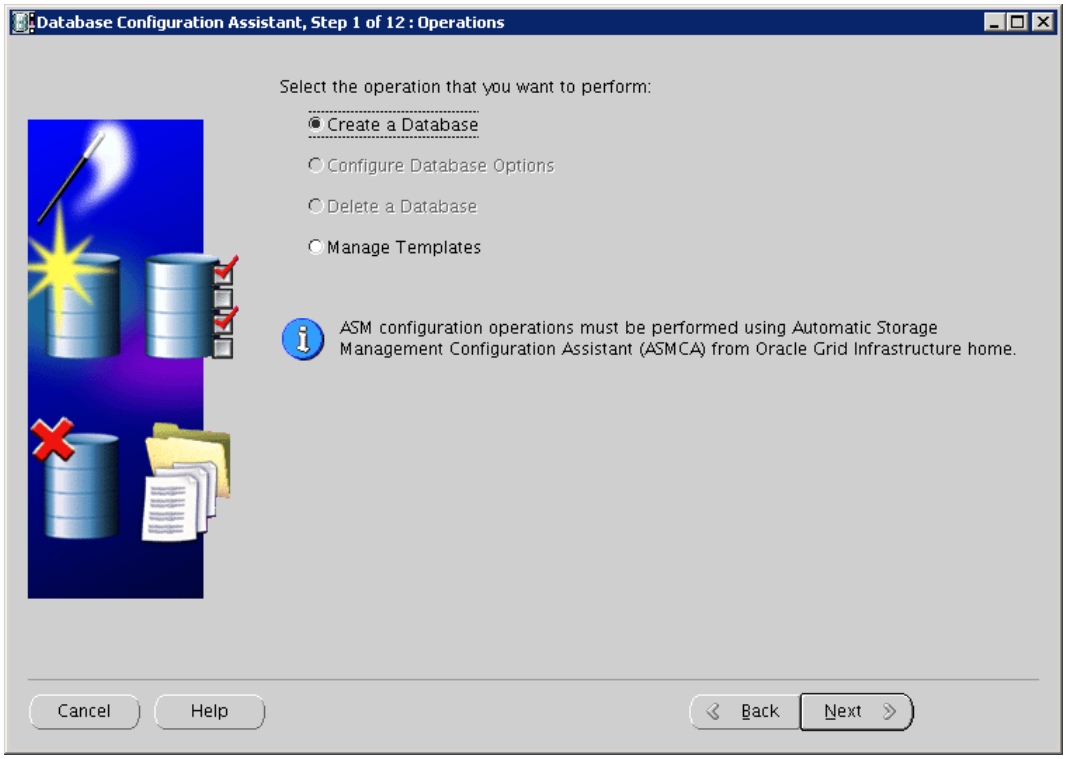
```
$> cd $ORACLE_HOME/bin
$> . export DISPLAY=<set DISPLAY>

$>./dbca
```

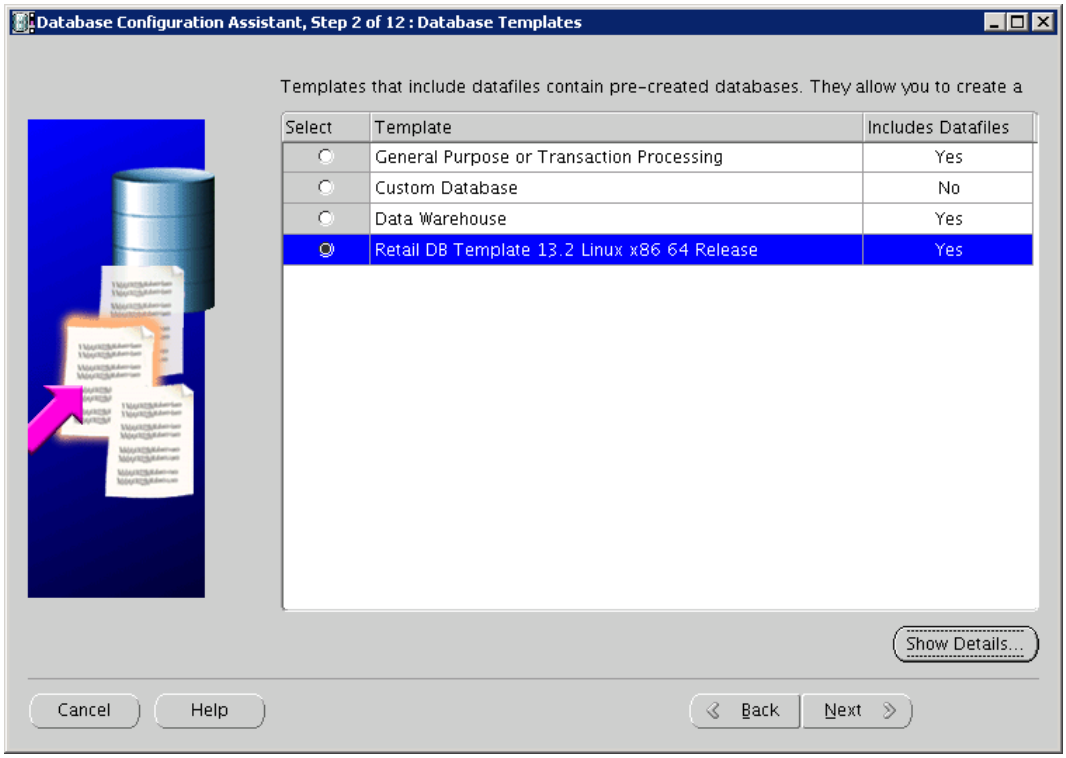
2. From the Welcome window, click **Next**.



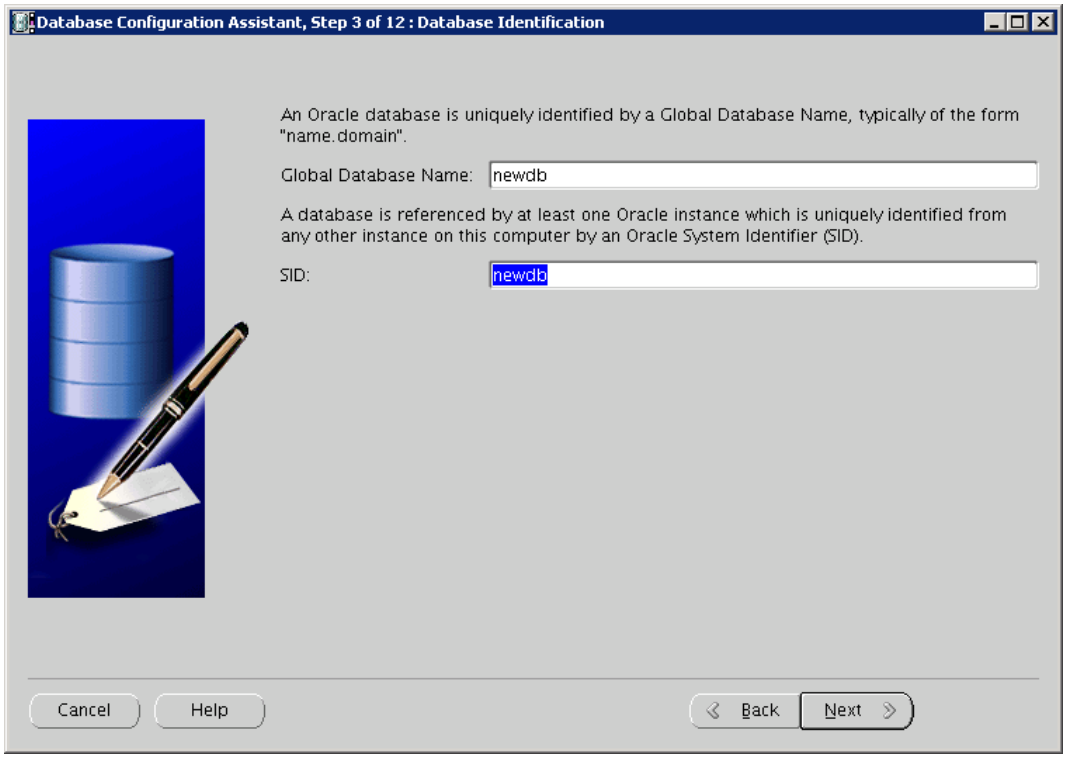
3. Select the Create Database option and click **Next**.



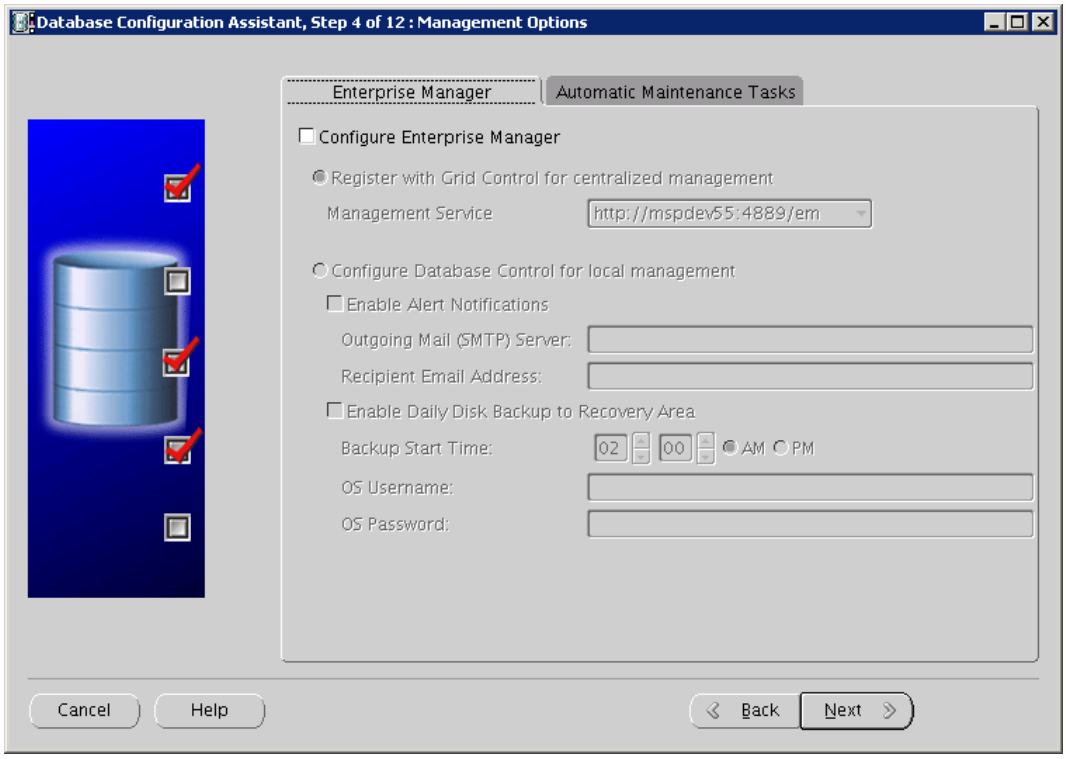
4. Select Retail DB Template 13.2 Linux x86_64 Release and click **Next**.



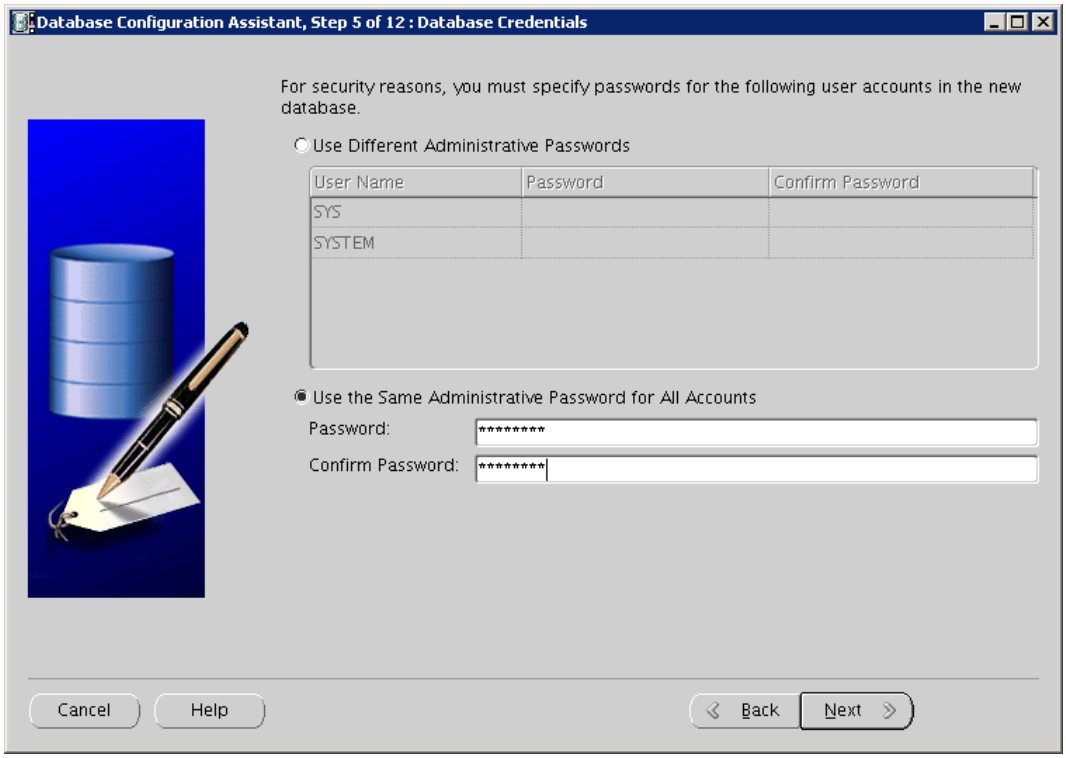
5. Enter the new database name in the Global Database Name and the SID in the SID field. Click **Next**.



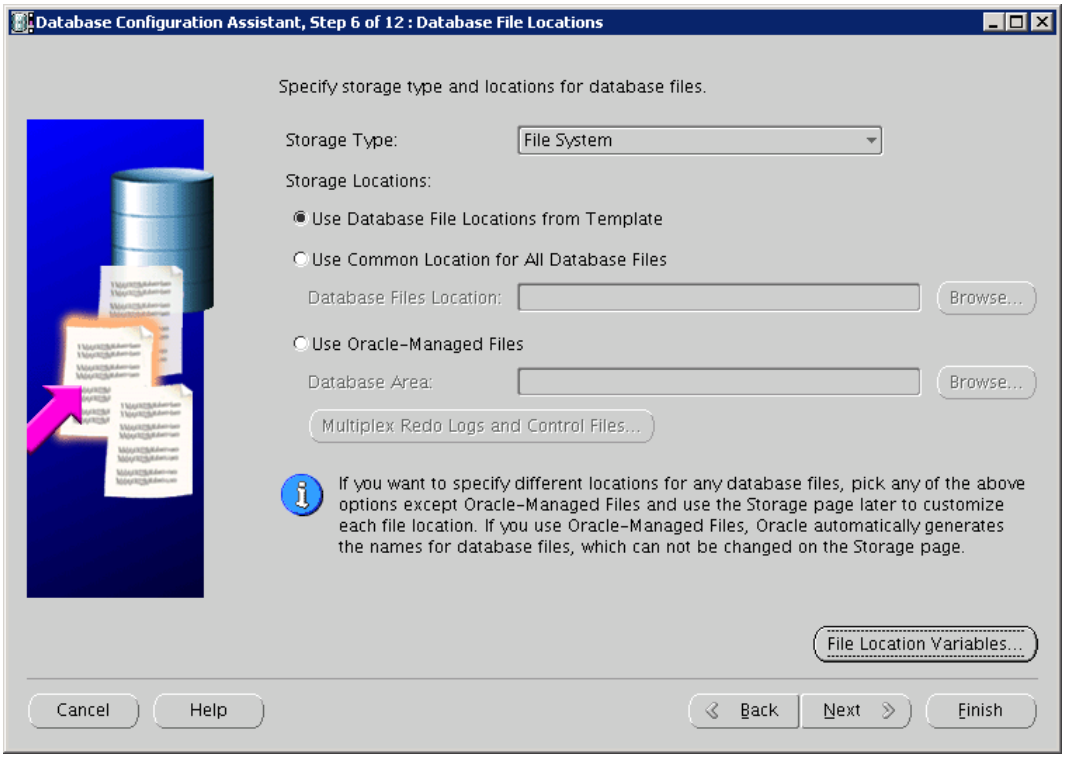
6. Deselect Configure Enterprise Manager if the option is checked and click **Next**.



7. Determine whether you will use the same passwords for both the SYS and the SYSTEM accounts and select the appropriate option. Enter the passwords for both accounts accordingly and click **Next**.

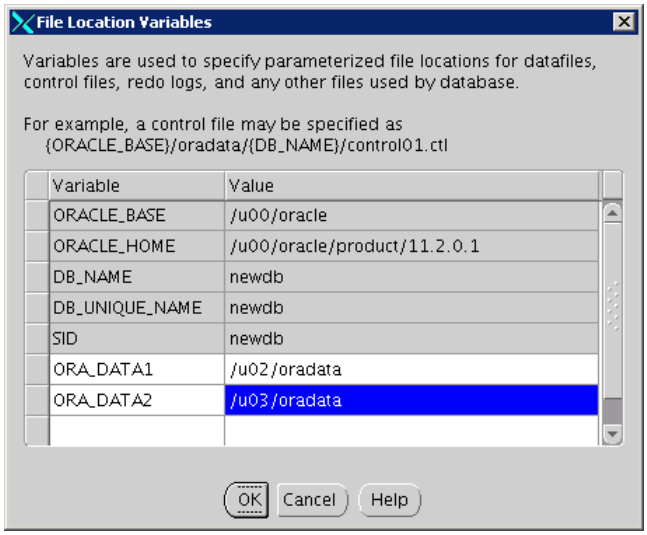


8. Select Use Database File Location from Template and click the File Location Variables button to bring up the File Location Variables screen.

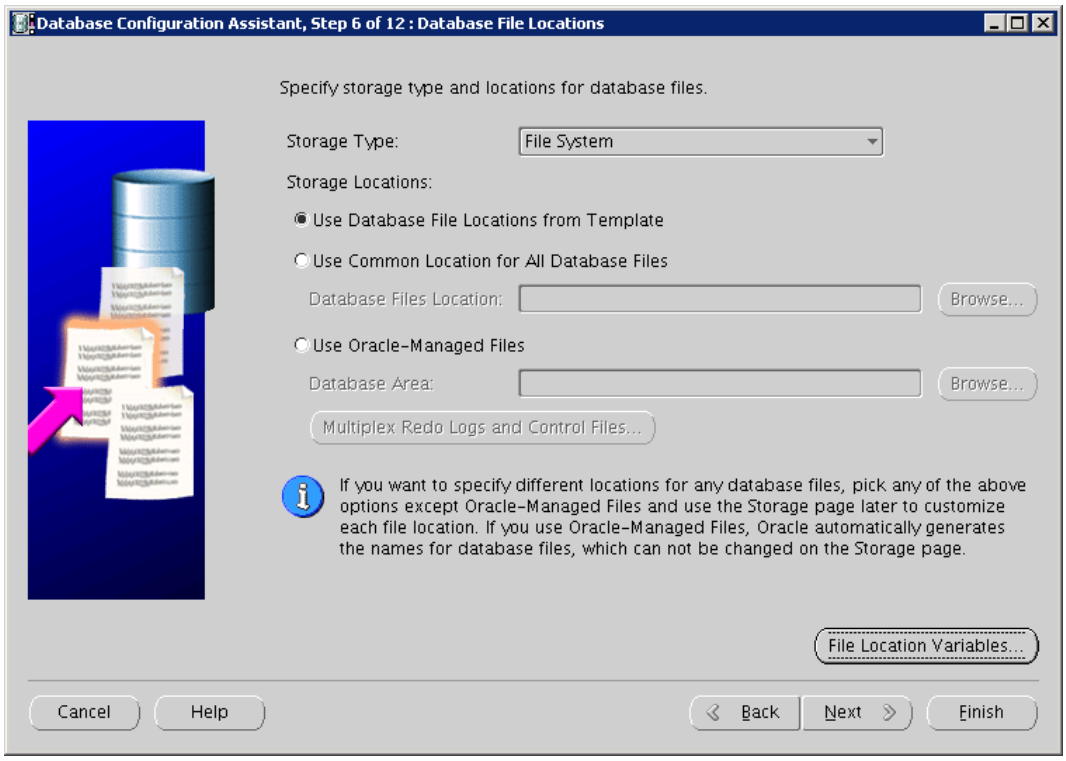


9. Verify that the values of ORACLE_BASE, ORACLE_HOME, DB_NAME, DB_UNIQUE_NAME and SID are set correctly, and then enter the locations of the datafiles in ORA_DATA1 and ORA_DATA2. ORA_DATA1 and ORA_DATA2 hold the values for the locations of datafiles, redolog files, and controlfiles.

10. Click **OK** to exit this screen. You are returned to the Database Files Location screen.

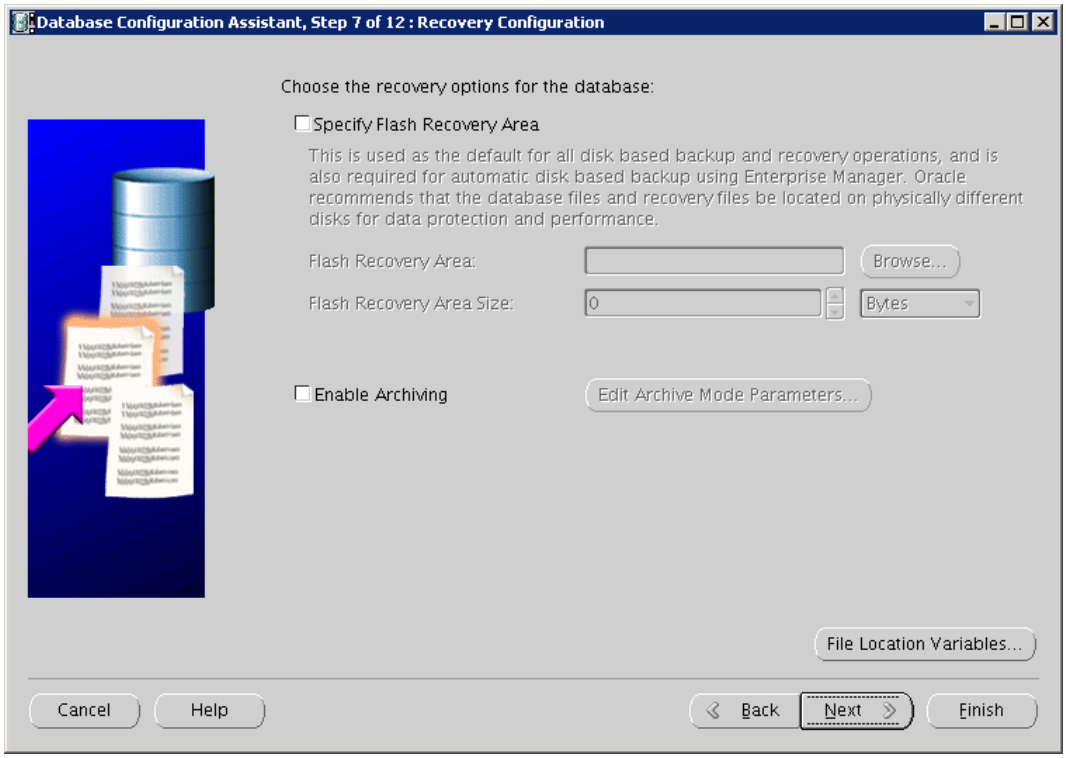


11. From the Database File Locations screen, click **Next**.



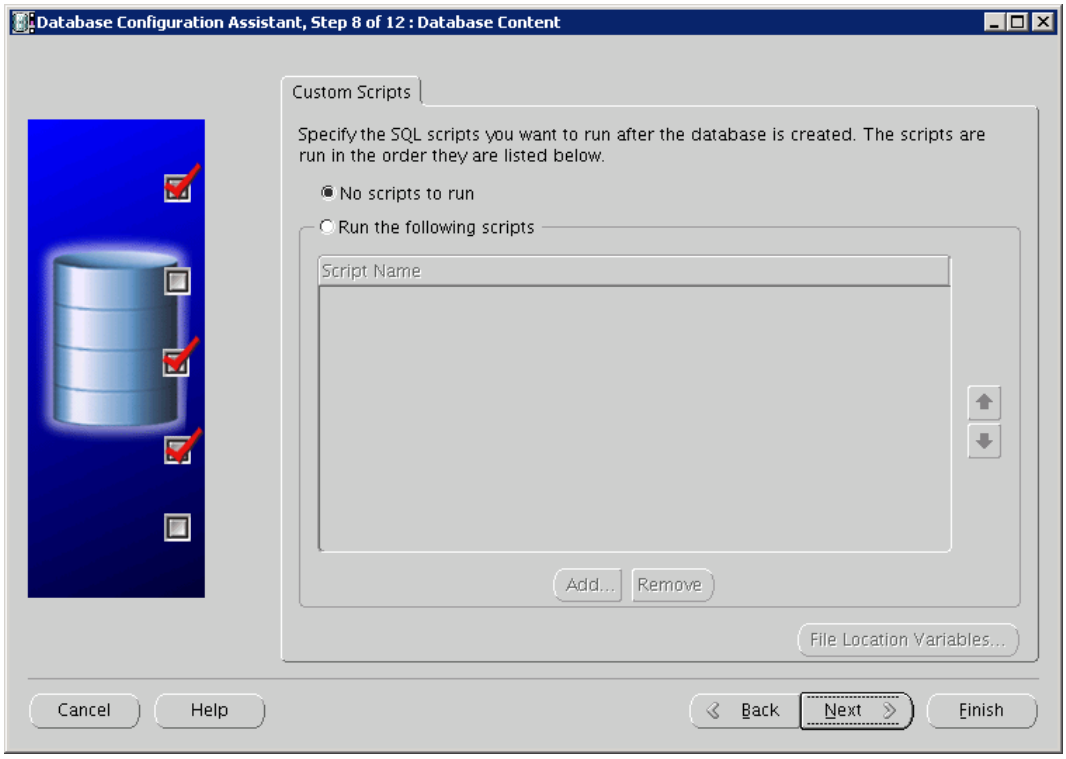
12. If you use flash recovery area, select **Specify Flash Recovery Area** and then enter values for the **Flash Recovery Area** and Flash Recovery Area Size felds.
13. Select **Enable Archiving** if the database is going to be in archived mode.

14. Click **Next**.



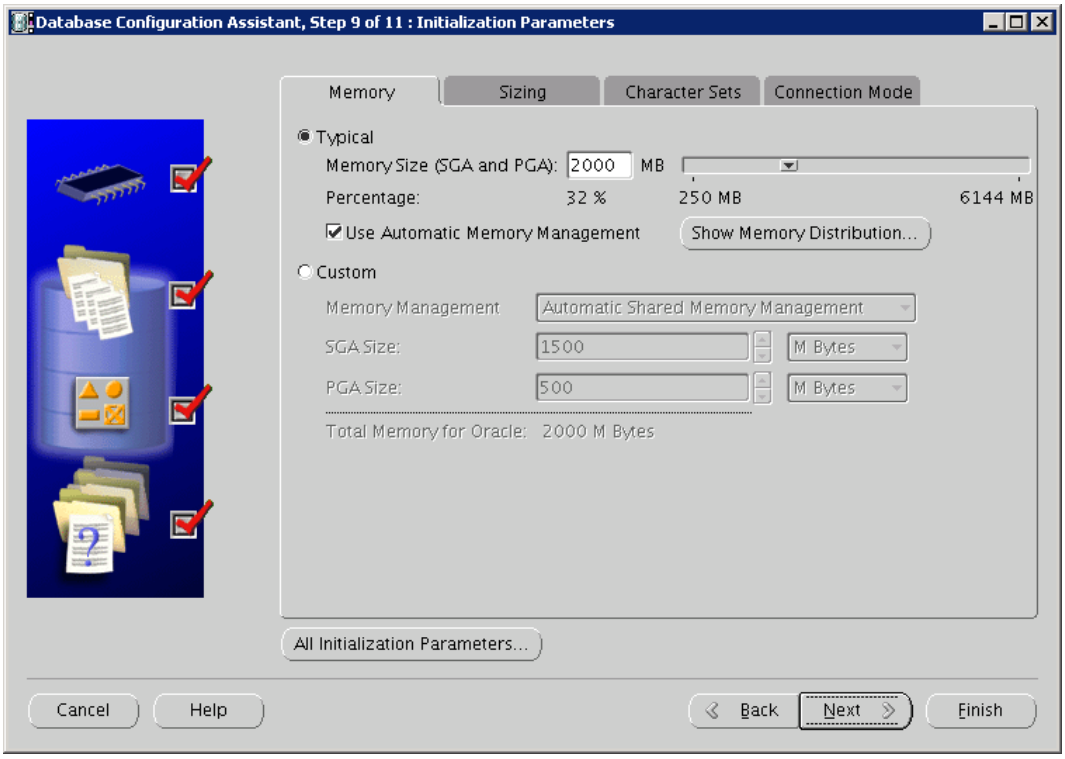
15. If there are any scripts to be run during the database creation process, select **Run the following scripts** and enter the names of the scripts. If no scripts are to be run, select **No scripts to run**.

16. Click **Next**.

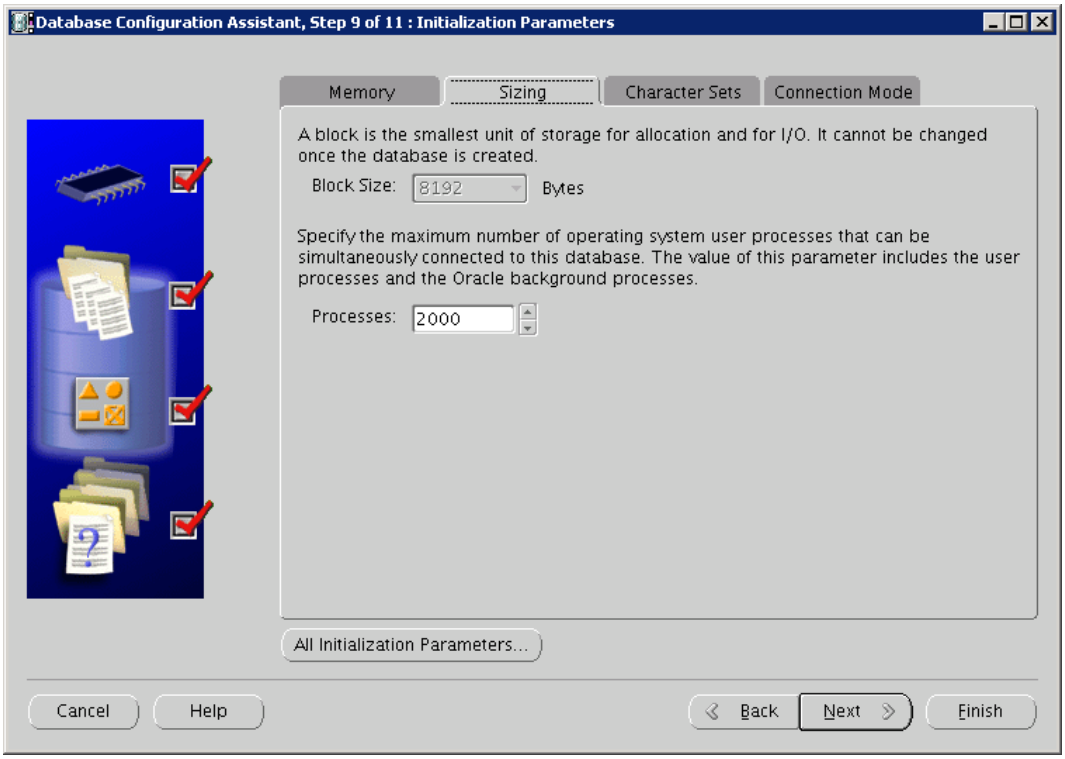


17. If the setting of Typical is acceptable, leave the field selected. If not, check the Custom field and enter the values in the SGA Size and PGA Size accordingly.

18. Click **Next**.

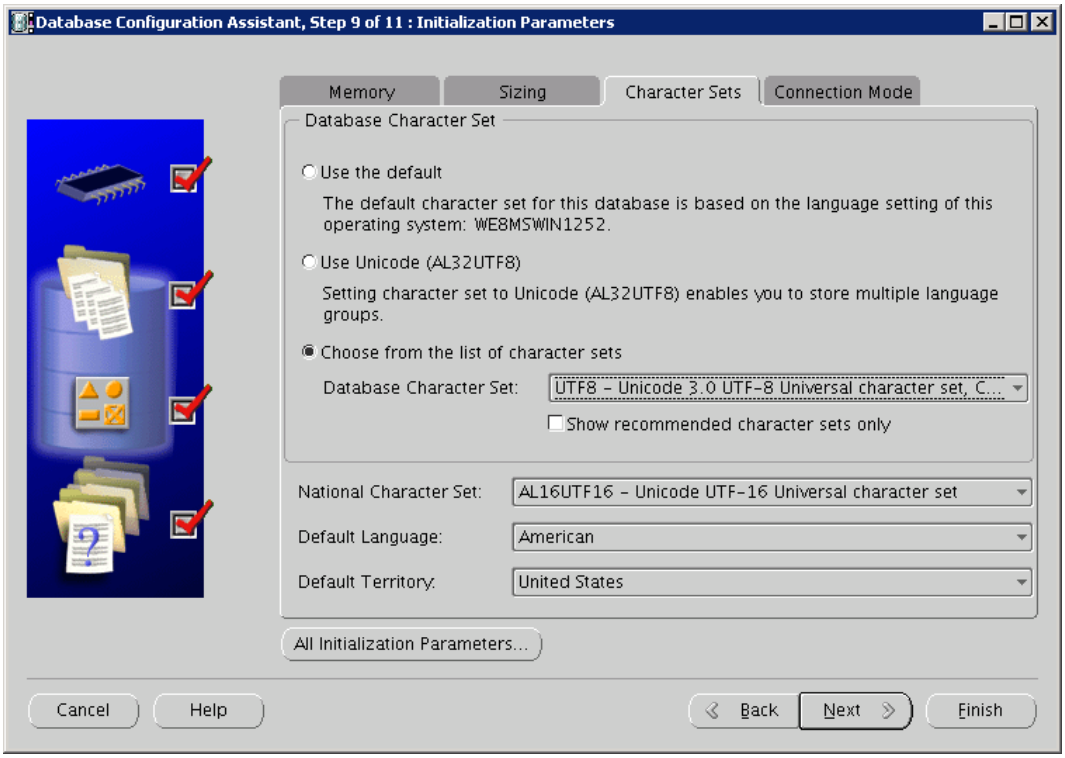


19. Adjust the values of Processes if necessary and click **Next**.

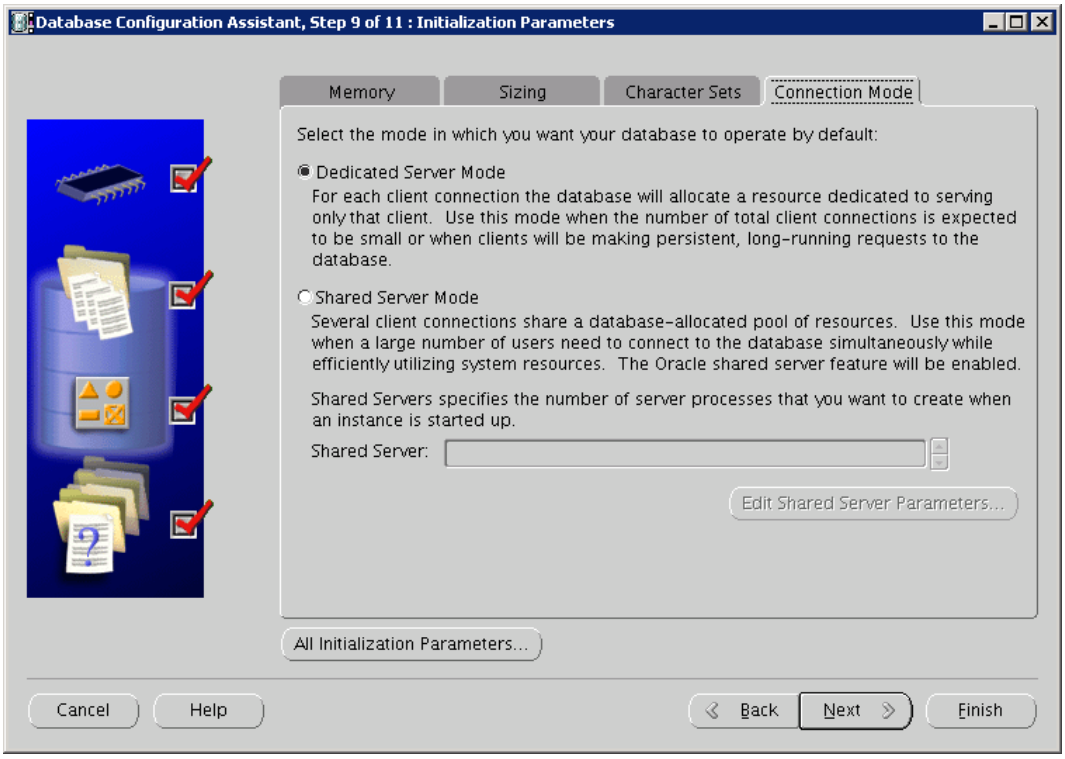


20. Uncheck Show recommended character set only' to bring up a complete list of character sets. Select **UTF8**.

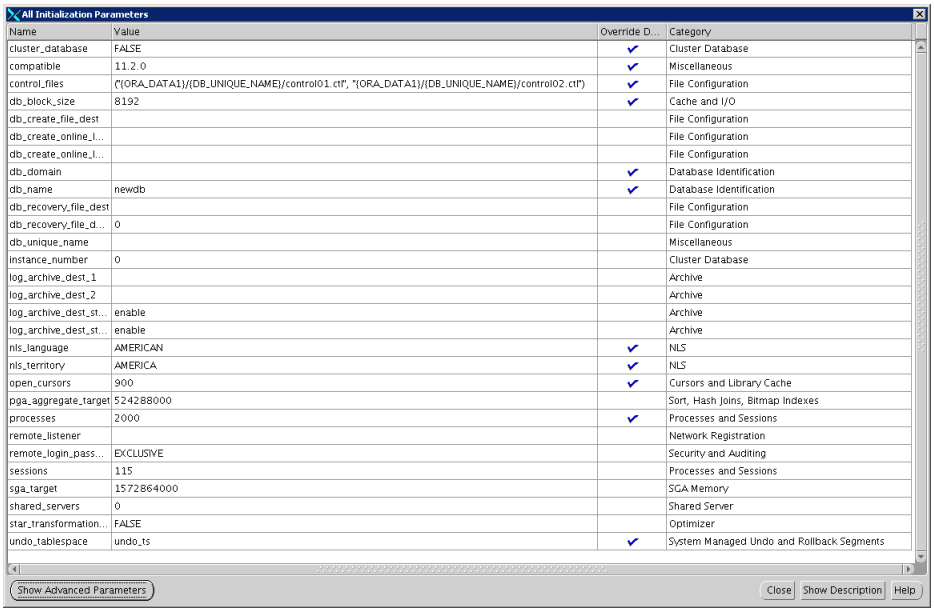
21. Click **Next**.



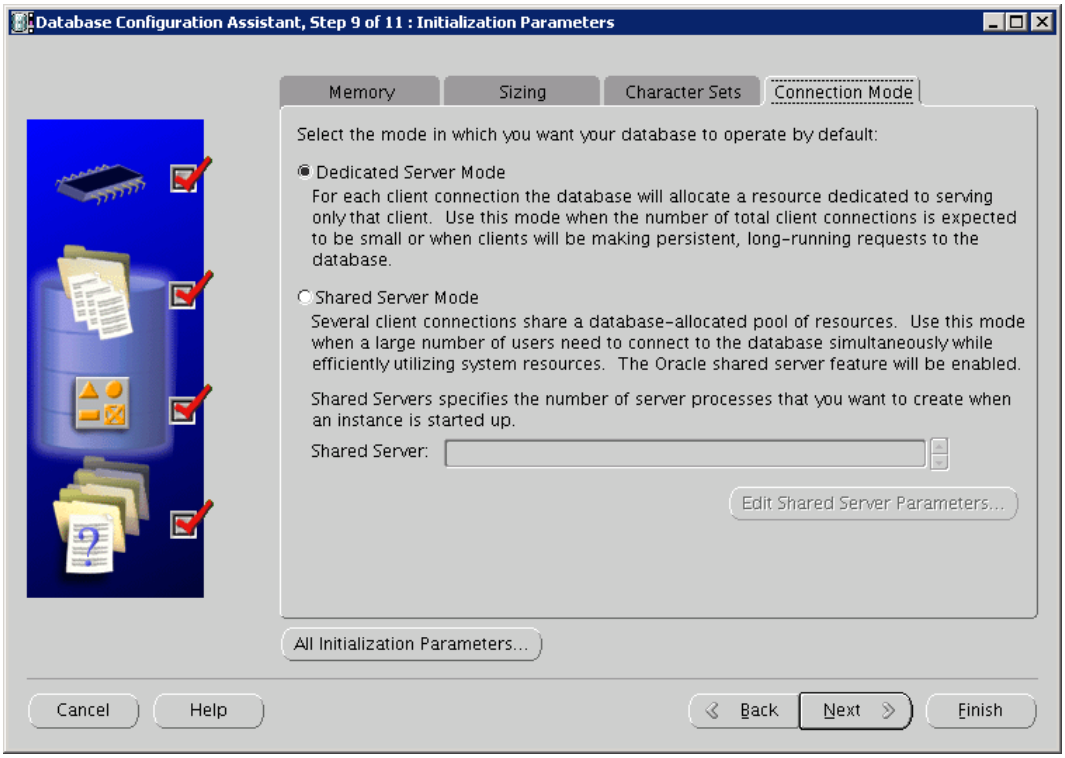
22. Select Dedicated Mode and click All Initialization Parameters to bring up a complete list of parameters.



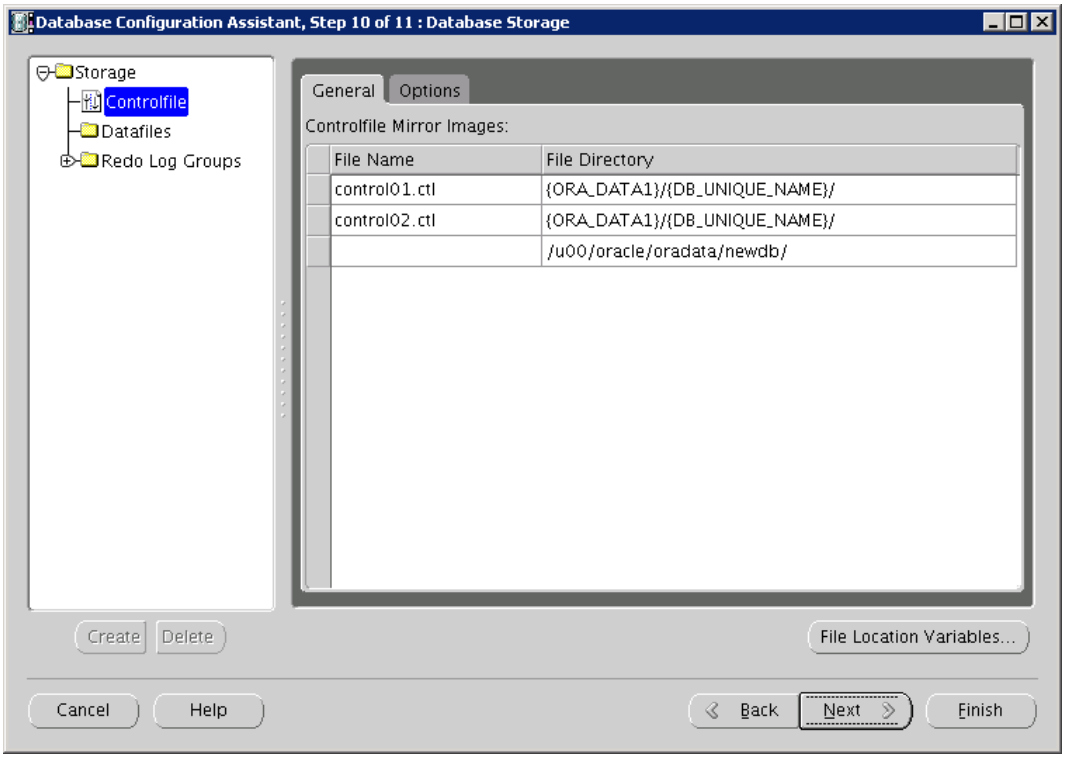
23. Select the values of the initialization parameters and adjust them if necessary. Click **Close** to exit.



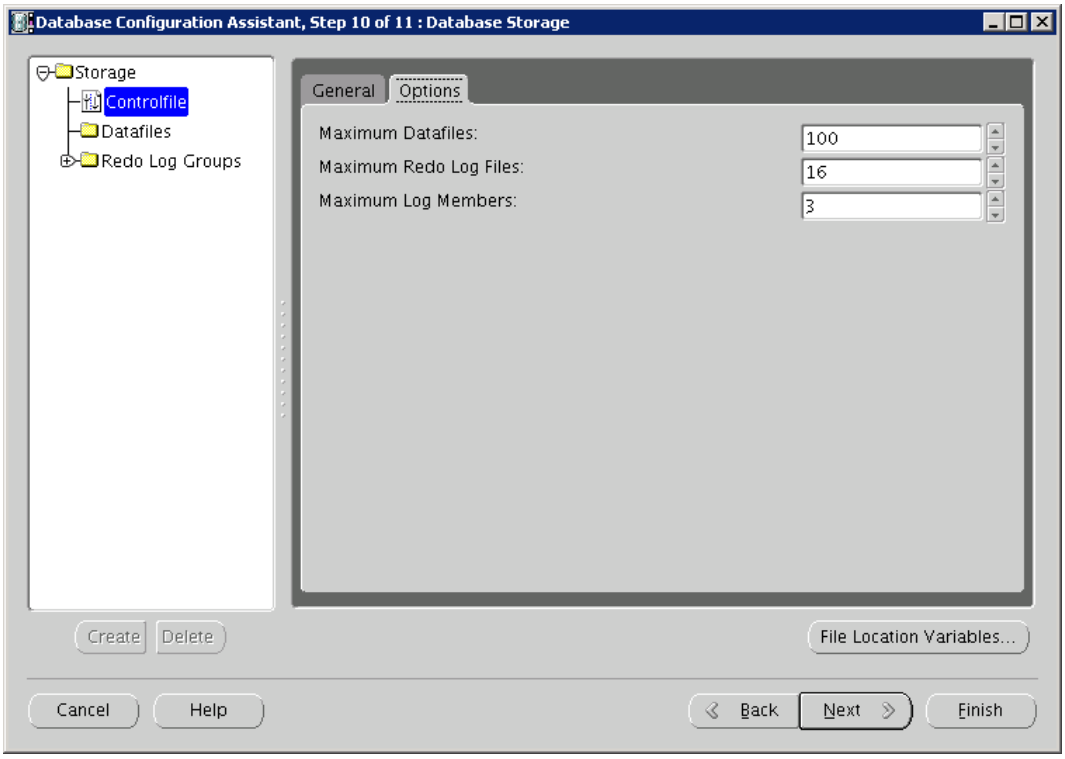
24. Ensure Dedicated Server Mode is checked then click **Next**.



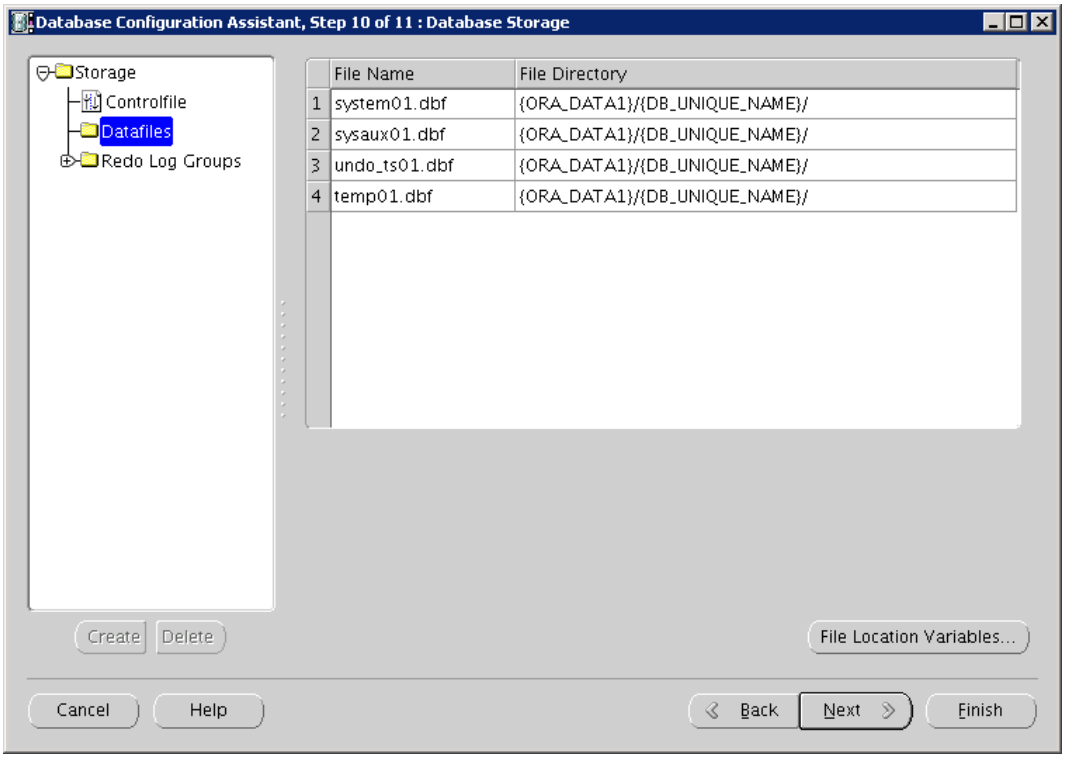
25. Click on the Controlfile tab in the left window and check the locations and the names of the control files.



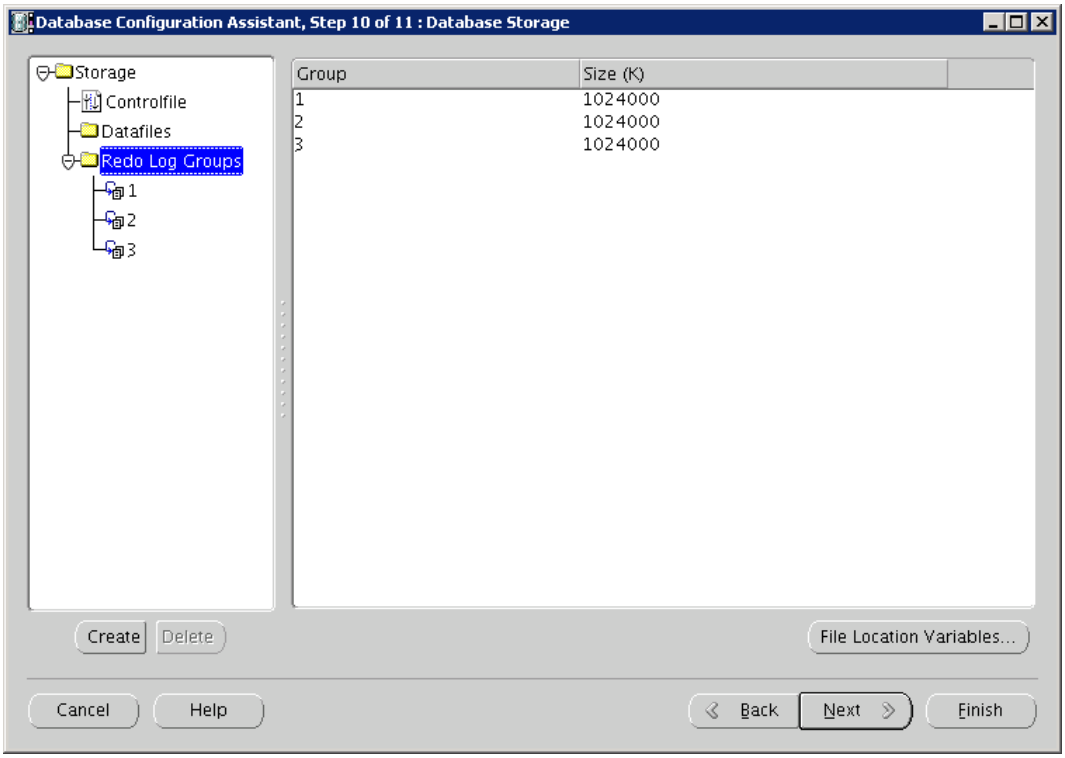
26. Click the Options tab to verify the values shown.



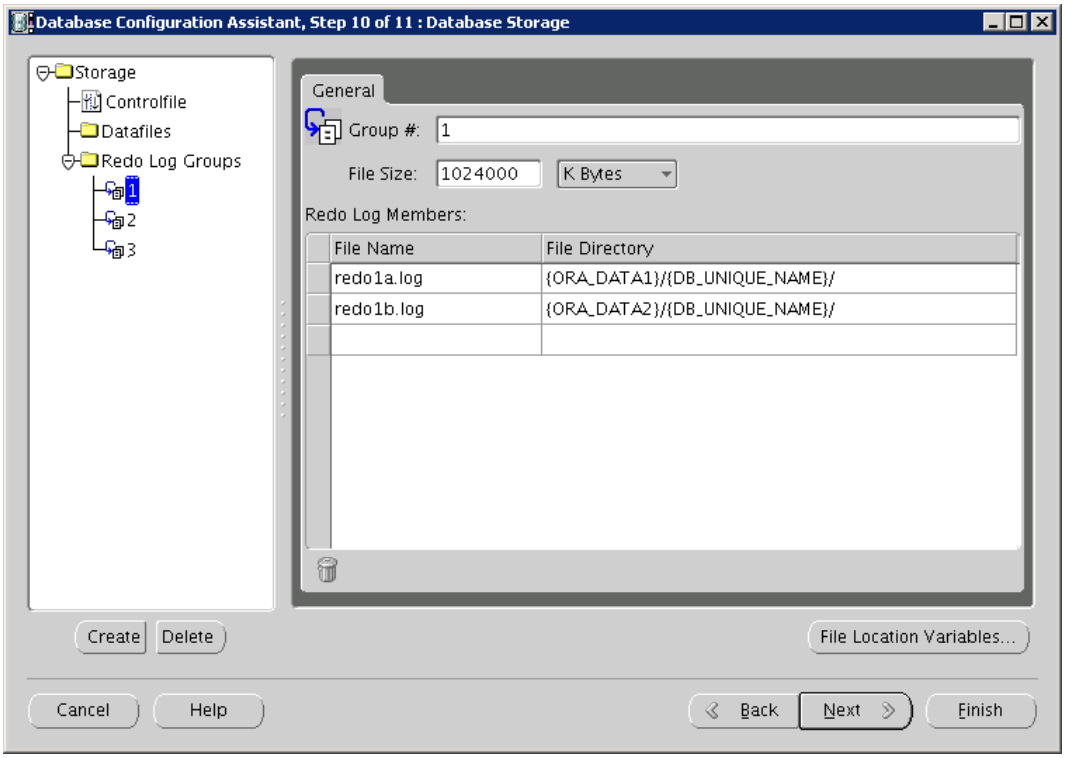
27. Click on Datafile tab on the left window and check the locations and the names of the datafiles.



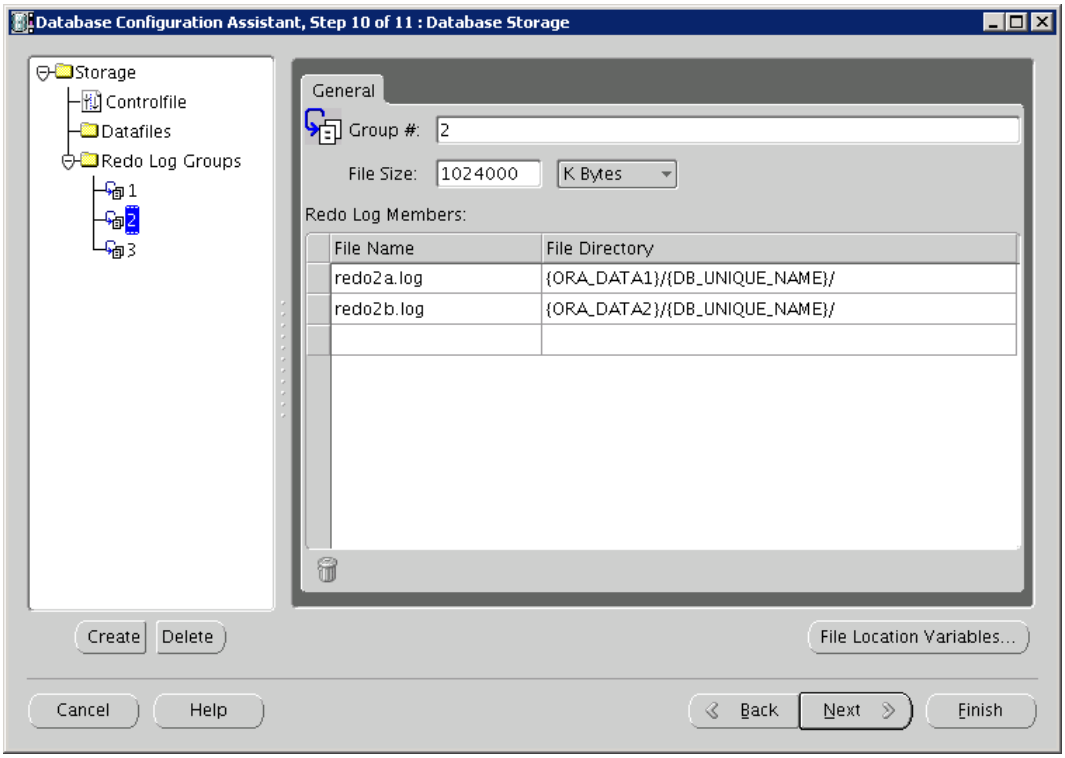
28. Click the Redo Log Groups tab on the left window and check the groups and their sizes.



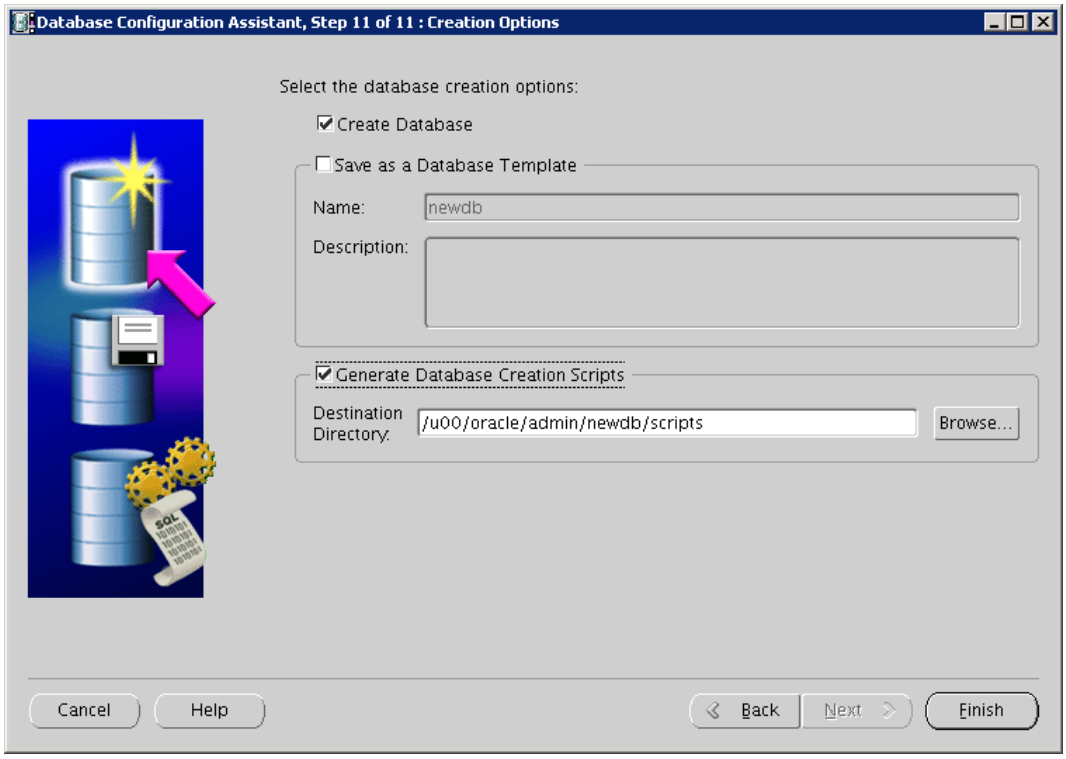
29. Click the Redo Log Groups 1 on the left window and verify the locations and names of the members.



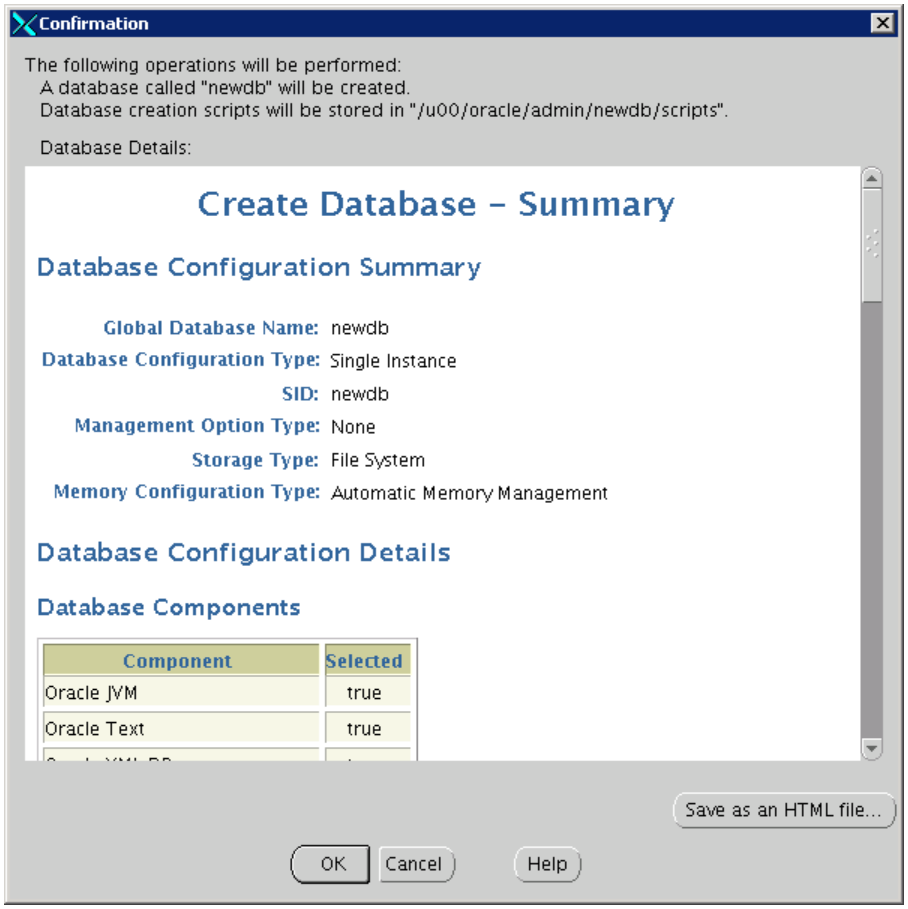
30. Click on the Redo Log Groups 2 and 3 on the left window and verify the locations and names of the members. Once all groups have been verified, click **Next**.



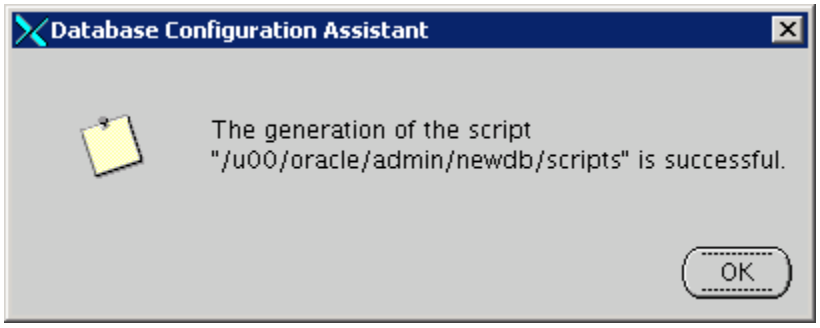
31. Select Create Database and Generate Database Creation Scripts. Cick **Finish**.



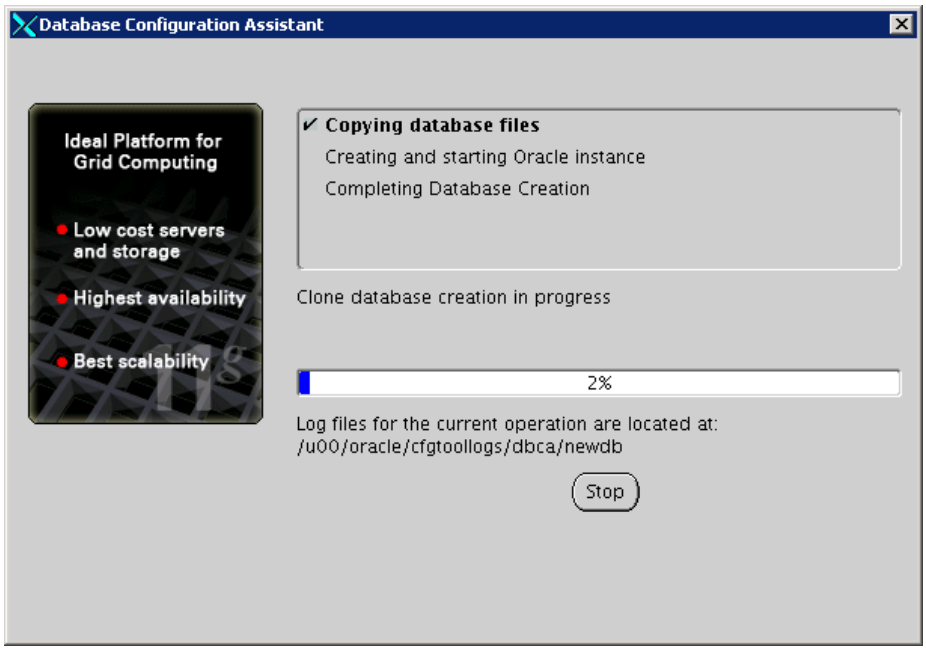
32. A summary page is shown, displaying all settings for the instance about to be created. Validate values for the last time. Cick **OK**.



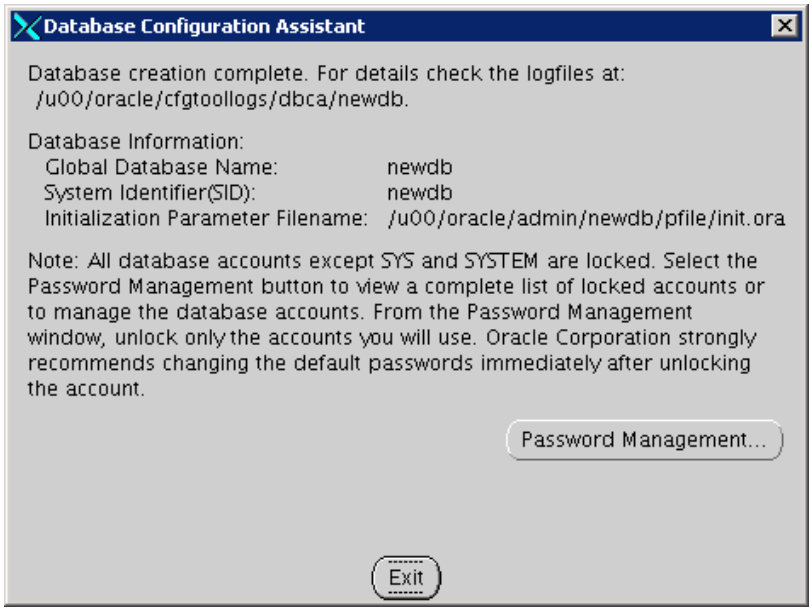
33. The database creation script is generated. Click **OK** to start the database creation process.



The Database creation in progress screen is displayed.

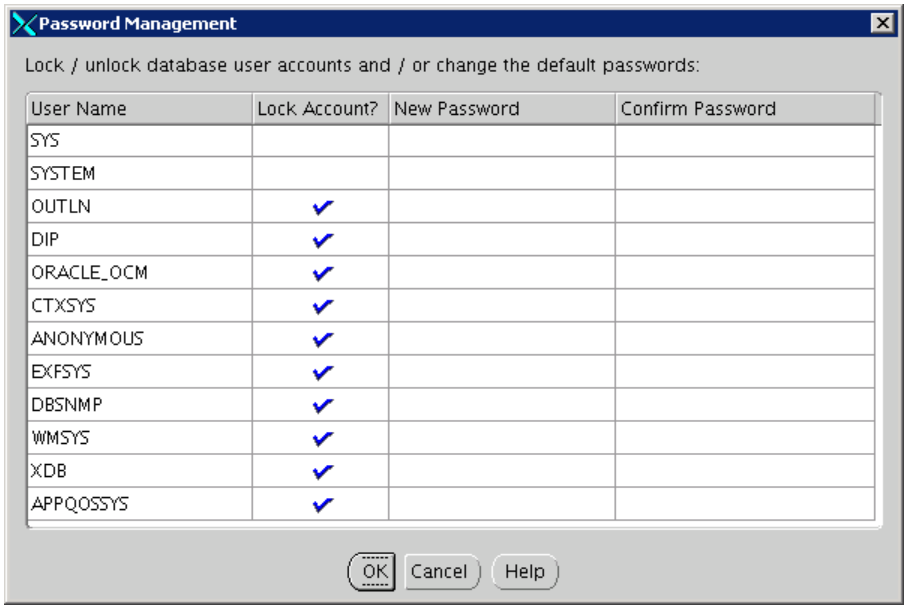


34. When the Database Configuration Assistant screen appears, click the Password Management button to lock/unlock the user accounts and manage their passwords.

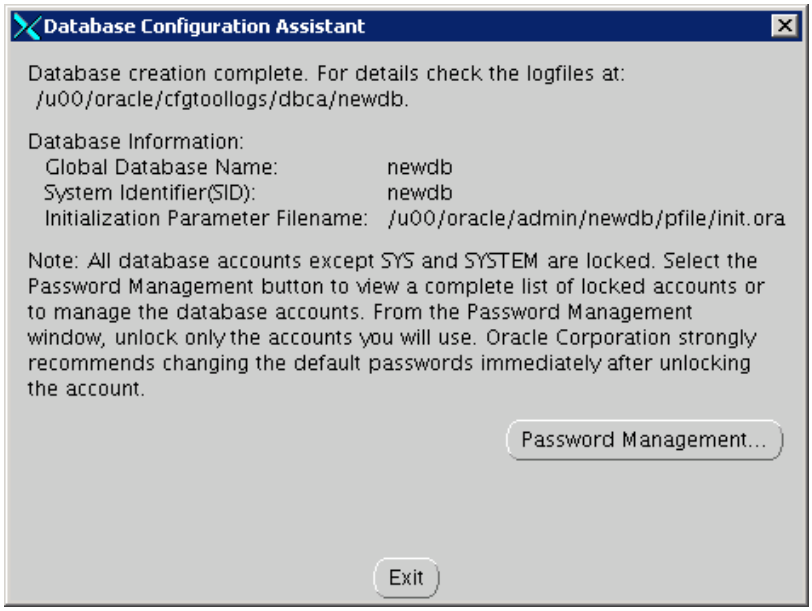


35. Lock/Unlock database user accounts accordingly. You may also set their passwords on this screen if you do not wish to keep the default passwords.

36. Click **OK** to close the dialog box. You are returned to the Database Configuration Assistant screen.



37. When the database creation process has completed, click **Exit**.



38. Configure the listener and create the tnsnames entry.

Silent Mode Instance Creation Using DBCA

1. Log in to UNIX as the oracle user; typically the user that owns the Oracle Database software.
2. Set values ORACLE_HOME for ORACLE_BASE.

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

3. Update/set the values of ORA_DATA1 and ORA_DATA2 in the file $ORACLE_HOME/assistants/dbca/templates /Retail_DB_Template_13.2_OS_Platform_Release_variables.txt so they hold the paths of the datafile locations as seen in the example below:

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

4. Execute dbca in silent mode to create the database.

```
cd $ORACLE_HOME/bin
./dbca -silent -createDatabase -templateName
Retail_DB_Template_13.2_Linux_x86_64_Release.dbc -gdbname <DB NAME> -sid <DB
NAME> -characterSet UTF8 -nationalCharacterSet AL16UTF16 -sysPassword <SYS
PASSWORD> -systemPassword <SYSTEM PASSWORD> -emConfiguration NONE -
variablesFile $ORACLE_HOME/assistants/dbca/templates/Retail_DB_Template_13.2_
OS_Platform _Release_variables.txt -continueOnNonFatalErrors true
```

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

5. A database instance is created.
6. Configure the listener and create the tnsnames entry.

Create the Tablespaces

Modify STAGING_DIR/rms/dbschema/create_db/create_rms_tablespaces.sql. Refer to Appendix C and the section below. Once this script has been modified, execute it in SQL*Plus as sys. Review create_rms_tablespaces.log for errors and correct as needed.

> **Note:** The partitioning strategy determines the size of RMS tablespaces. Be aware that increasing the number of partitions may necessitate an increase in the size of the required tablespaces. It is important to be accurate when sizing tablespaces prior to the installation of RMS. Failure to do so results in "insufficient space" errors which require a complete re-install of RMS.

The STAGING_DIR/rms/dbschema/create_db/create_rms_tablespaces.sql script contains the DDL for creating the required tablespaces which can extend up to the following sizes:

TABLESPACE_NAME	SIZE
RETEK_INDEX	12G
RETEK_DATA	6G
LOB_DATA	2G
USERS	2G

These sizes are sufficient if the initial values in the STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/RMS_partition_definition.xls spreadsheet are used without modifications. Although using the initial values is not recommended for a production environment, it is possible to use them for the purpose of creating a small test environment. For additional assistance with production database sizing, contact Oracle Retail Services.

Create the Schema Owner for RMS

Create an Oracle schema that will own the RMS application. Refer to Appendix: RMS User Creation Script and the section below.

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

1. Change directories to STAGING_DIR/rms/dbschema/create_db
2. The create_user script relies on an empty role, developer, being created. Log into sqlplus as sysdba and run the following command to create that role.

   ```
   SQL> create role developer;
   ```

3. Enter the following command to create the schema owner.

   ```
   SQL> @create_user.sql
   ```

   The following prompts will occur:

   - Schema Owner – the Oracle user that will own all RMS objects. Referred to in this install guide as RMS13DEV
   - Password – the password for RMS13DEV
   - Temp Tablespace – the temporary tablespace for RMS13DEV

4. Check the log file create_<Schema Owner>.lst for any errors. This log file should be removed to prevent the password from being compromised.

Create the Database User for Allocation

To create the database user for allocation, complete the following steps.

1. If Allocation will be installed using the RMS database schema installer, an additional database user is required for the Allocation temporary tables. Follow the same instructions as in the Create the Schema Owner for RMS section above to create this additional user, except use the create_user_generic.sql script located here:
2. STAGING_DIR/rms/dbschema/create_db/create_user_generic.sql

> **Example:** ALLOC13DEV

Review RIB CLOB Settings

1. The RMS database schema installer runs the RIB objects into the RMS schema. There are some RIB settings passed to the RIB CLOB (Character Large Object) scripts that you can configure ahead of time. Review the rms_rib_install.properties file for the settings passed to the RIB CLOB scripts by the RMS installer.
2. For more information on the RIB objects see the RIB documentation.

Run the RMS Database Schema Installer

> **Note:** See Appendix: RMS Database Schema Installer Screens for details on every screen and field in the database schema installer.

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

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

Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

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

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

Variable	Description	Example
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 L: Common Installation Errors for more details.
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. Depending on system resources, a typical installation can take 7-9 hours.
7. The RMS 13.2 Database Schema Installer provides the option of installing the Invoice Matching (ReIM) and Allocation database objects in addition to the RMS objects.

For the initial RMS 13.2 installation select the **Full** option on the **Full Install or Patch Option** screen. RMS 13.2.x patches released after RMS 13.2 will utilize the Patch option.

> **Note**: For Brazilian retailers planning to install ORFM/RMS Brazil Localization, this installation of RMS must include the following specific installation settings:

Primary Country = “BRAZIL (BR)”

Primary Currency = “Brazil Real (BRL)”

Uncheck “Enable VAT”

8. After the installer is complete, you can check its log file: rms-install-dbschema.<timestamp>.log. A .dbhistory file is created with a listing of all of the sql scripts that were run by the installer. A .dberrors file is created if any errors are encountered.
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`

Values to Remember for the Batch and Application Installers

After it has completed the schema installation, the installer prints some database settings that you need for the batch and application installers. These settings are also written to the end of the installer log file (rms-install-dbschema.<time>.log). Record these settings for use during the batch and application installations.

Resolving Errors Encountered During Database Schema Installation

If the database schema installer encounters any errors, it halts execution immediately and prints to the screen which SQL script it was running when the error occurred. It may also write the path to this script to the .dberrors file. When this happens, you must run that particular script using sqlplus. After you are able to complete execution of the script, delete the .dberrors file if it exists and run the installer again. You can run the installer in silent mode so that you do not have to go through the installer screens again. See Appendix: Installer Silent Mode in this document for instructions on silent mode.

See Appendix: Common Installation Errors in this document for a list of common installation errors.

Subsequent executions of the installer skip the SQL scripts which have already been executed in previous installer runs. This is possible because the installer maintains a .dbhistory file with a listing of the SQL scripts that have been run, and a set of *.processed files that indicate files that have successfully run. If you have dropped the RMS schema and want to start with a clean install, you can rerun the installer and choose to delete the .dbhistory and *.processed file so that the installer runs through all of the scripts again. If continuing an installation, it is recommended that you allow the installer to skip the files that it has already run.

Set Up Additional RMS Users

1. Additional users to the RMS application can be set up by following the same instructions as in the Create the Schema Owner for RMS section above, except use the create_user_generic.sql script located here:
STAGING_DIR/rms/dbschema/create_db/create_user_generic.sql

Note: Evaluate the use of multiple roles and assign appropriately to users, based on user responsibilities.

2. After users are set up, create synonyms to the owner schema for all tables, views, sequences, functions, procedures, packages and types to which the user has access.

3. Run the following scripts as the new user to give new users security privileges. These scripts can be found in the RMS database schema installer package under rms/dbschema/dbscripts_rms/utility.

```
SQL> @englishUser.sql
SQL> @superUser.sql
```

4. Grant the following privileges to the new user
   - grant dequeue any queue to <userid>;
   - grant enqueue any queue to <userid>;
   - grant manage any queue to <userid>;

2

Batch Installation Tasks—Full

This section includes steps for batch installation.

Create Staging Directory for RMS Batch Files

To create a staging directory for RMS batch files, complete the following steps.

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS batch installation software or use the same staging directory as created in the database schema step above. There should be a minimum of 35 MB disk space available in this location.
3. Copy the rms13batch.zip file from the RMS 13.2 release to the staging directory. This is referred to as STAGING_DIR when installing the RMS batch software.
4. Change directories to STAGING_DIR and extract the rms13batch.zip file. This creates an rms/batch subdirectory under STAGING_DIR.

Run the RMS Batch Installer

Note: See contains details on every screen and field in the batch installer.

1. Change directories to STAGING_DIR/rms/batch. This directory was created when the rms13batch.zip file was expanded under STAGING_DIR.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc)

   Example: prompt$. oraenv

   ORACLE_SID = [] ? mydb

   prompt$

   Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

   Example: prompt$ echo $ORACLE_HOME

   /u00/oracle/product/mydbversion

   prompt$ echo $ORACLE_SID

   mydb

3. Verify that the following executables are available from PATH: make, makedepend, cc, ar.

   Example: Here are some locations where makedepend is commonly found:

   Linux: /usr/X11R6/bin
   AIX: /usr/X11R6/bin

4. 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
DISPLAY	Address and port of X server on desktop system of user running install. Optional for batch installer	DISPLAY=<IP address>:0 export DISPLAY

5. 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 L: Common Installation Errors for more details.
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]

   Depending on system resources, a typical RMS batch installation takes anywhere from 20 to 60 minutes.

   The installer will ask for an installation directory. This is the destination directory for the RMS 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.
7. After the installer is complete, you can check its log file: rms.batch.install.<timestamp>.log.
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`

Resolving Errors Encountered During Batch Installation

The RMS batch installer is a full install that starts from the beginning each time it is run. If you encounter errors in your environment, after resolving the issue you can safely run the batch installer again to attempt another installation.

RETL

The RMS batch installer installs the RETL files under INSTALL_DIR.

See Appendix: RMS RETL Instructions in this document for more information about RETL.

Data Conversion Scripts

To install data conversion scripts, complete the following steps.

1. The RMS batch installer installs the data conversion scripts under INSTALL_DIR/external/scripts. To complete the setup of these files, perform the following steps.
2. Create the following new directories:
3. INSTALL_DIR/external/data
4. INSTALL_DIR/external/logs
5. The RMS Batch installer should have already created INSTALL_DIR/scripts.
6. Log into sqlplus as SYSTEM and run the following commands:

```
SQL> create or replace directory rms13dev_ext_data as
'INSTALL_DIR/external/data';
SQL> create or replace directory rms13dev_ext_logs as
'INSTALL_DIR/external/logs';
```

> **Note:** You need to replace INSTALL_DIR with your INSTALL_DIR and you can rename the external data and log directory.
>
> **Note:** The user that creates these directories owns them.
>
> **Note:** The data and logs directories should be chmoded 777.

7. Log into sqlplus as SYSTEM and grant access to them:
8. Grant the following privileges to any other users who will be using data conversion.

```
SQL> grant read on directory rms13dev_ext_data to RMS13DEVc;
SQL> grant read, write on directory rms13dev_ext_logs to RMS13DEV;
```

3

Application Server Installation Tasks—Full

It is assumed that WebLogic 11g version 10.3.3 (WLS) with Forms 11.1.1.3 has already been installed. If not, refer to "Check Application Server Requirements" in Chapter 1, "Preinstallation Tasks" before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in "Create Staging Directory for RMS Application Files" in Chapter 1.

In order to use WebLogic for manual compilation of RMS 13 forms modules, Oracle Forms Services 11g Release 1 (11.1.1.3) must be used. Please refer to the Oracle Forms Services 11g Release 1 (11.1.1.3) documentation for the steps to manually compile objects.

> **Note:** It is necessary to have $ORACLE_HOME/network/admin/tnsnames.ora file configured in this WLS installation. Forms will use this information for connectivity. Refer to Appendix B for an example setup of the tnsnames.ora file.

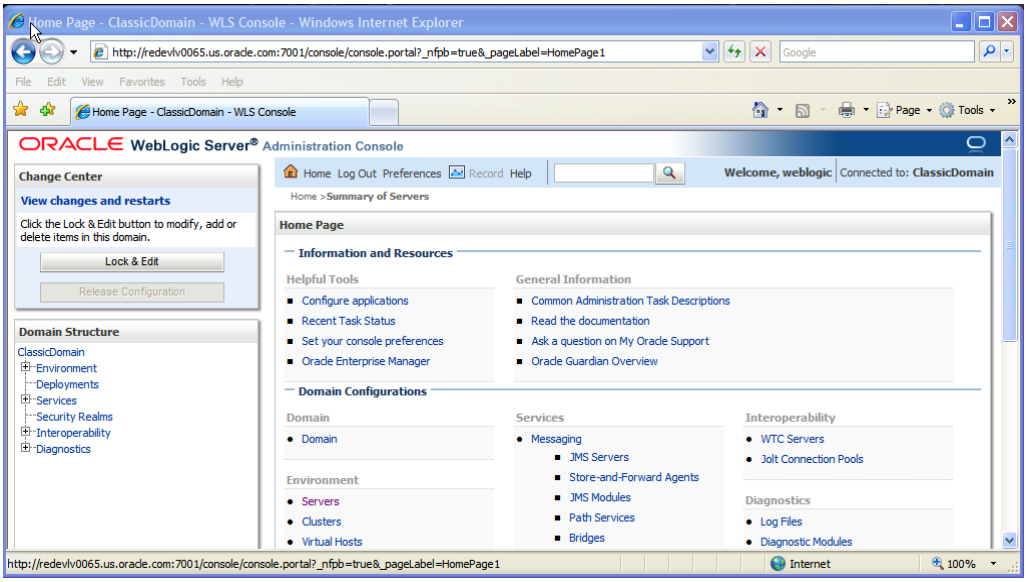
Prepare Application Server for RMS

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

1. The Tk2Motif.rgb file that is sent out with WebLogic (10.3.3) must be modified. It located at the following location:$ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
2. Make a copy of the file Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).
3. Modify the file Tk2Motif.rgb file so that it contains the following line:
   ```
   Tk2Motif*fontMapCs: iso8859-2=UTF8
   ```
4. Copy $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/Tk2Motif.rgb to $ORACLE_HOME/guicommon/tk/admin/Tk2Motif.rgb

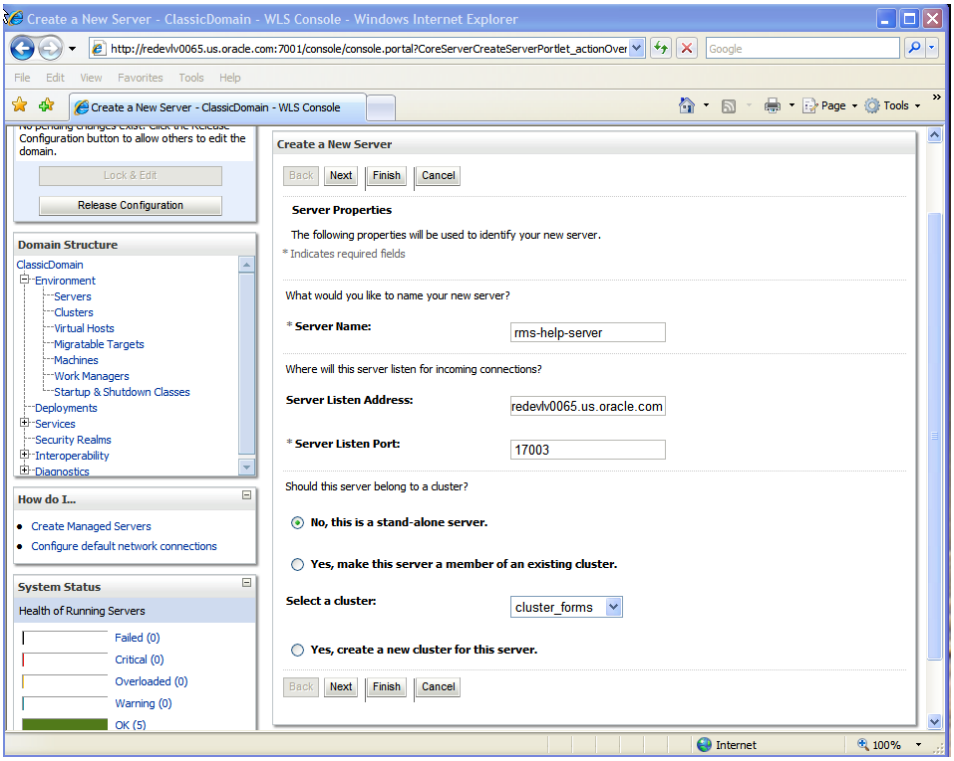
Create RMS Help Managed Server

1. Log in to the admin console.



2. Click **Lock & Edit**.

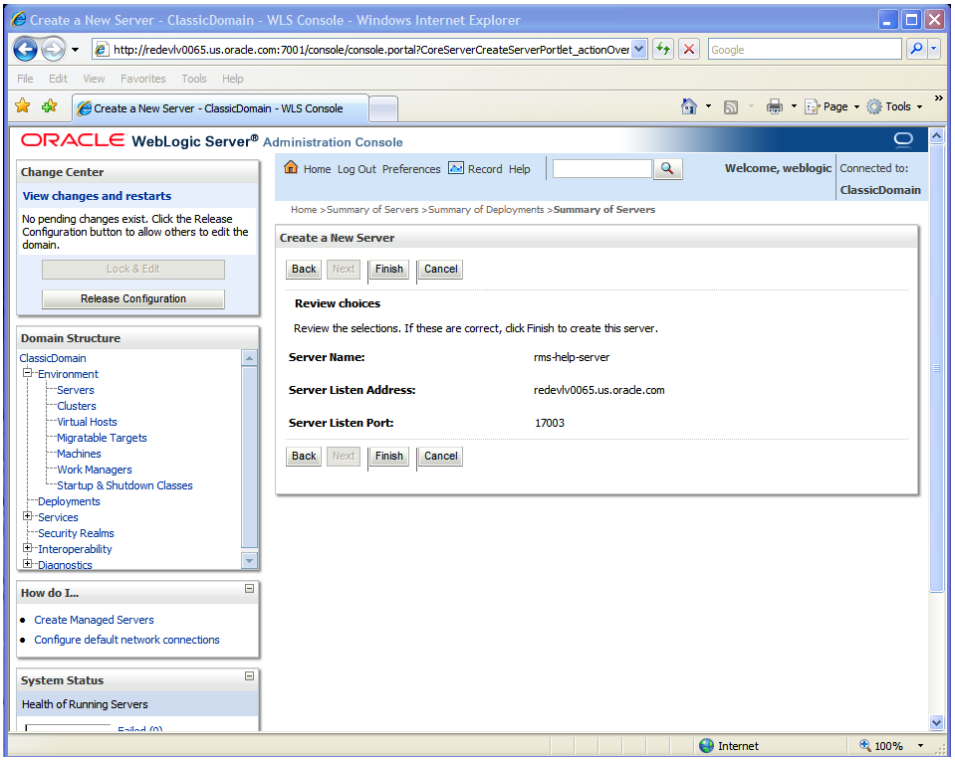
3. Navigate to Environment > Servers and select new tab of the servers on the right side.



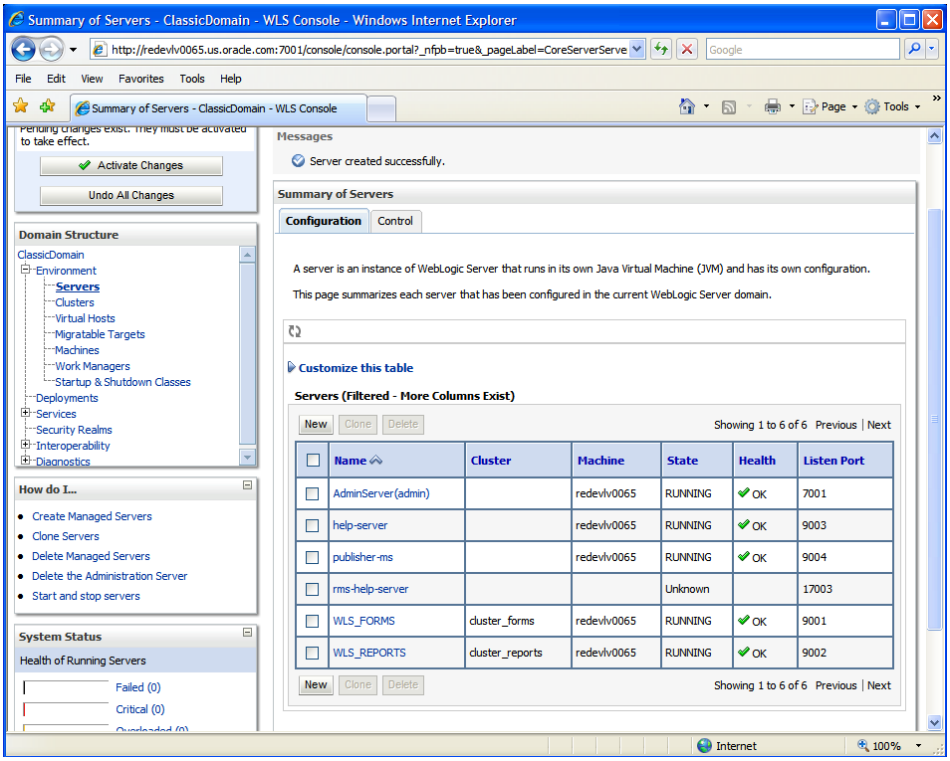
4. Set the following variables:
   - **Server Name**: These should be some name specific to your application targeted (for example, rms-help-server).
   - **Server Listen Address**: <weblogic server> (ie redevlv0065.us.oracle.com)
   - **Server Listen Port**: A free port; check for availability.

     A suggestion is to increment the AdminServer port by two and keep incrementing by two for each managed server (for example, 17003, 17005, 17007, and so on).

5. Click **Next**.


Create a New Server - ClassicDomain - WLS Console - Windows Internet Explorer
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
Health of Running Servers
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to: ClassicDomain
Home >Summary of Servers >Summary of Deployments >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: rms-help-server
Server Listen Address: redevlv0065.us.oracle.com
Server Listen Port: 17003
Back Next Finish Cancel

6. Click **Finish**.

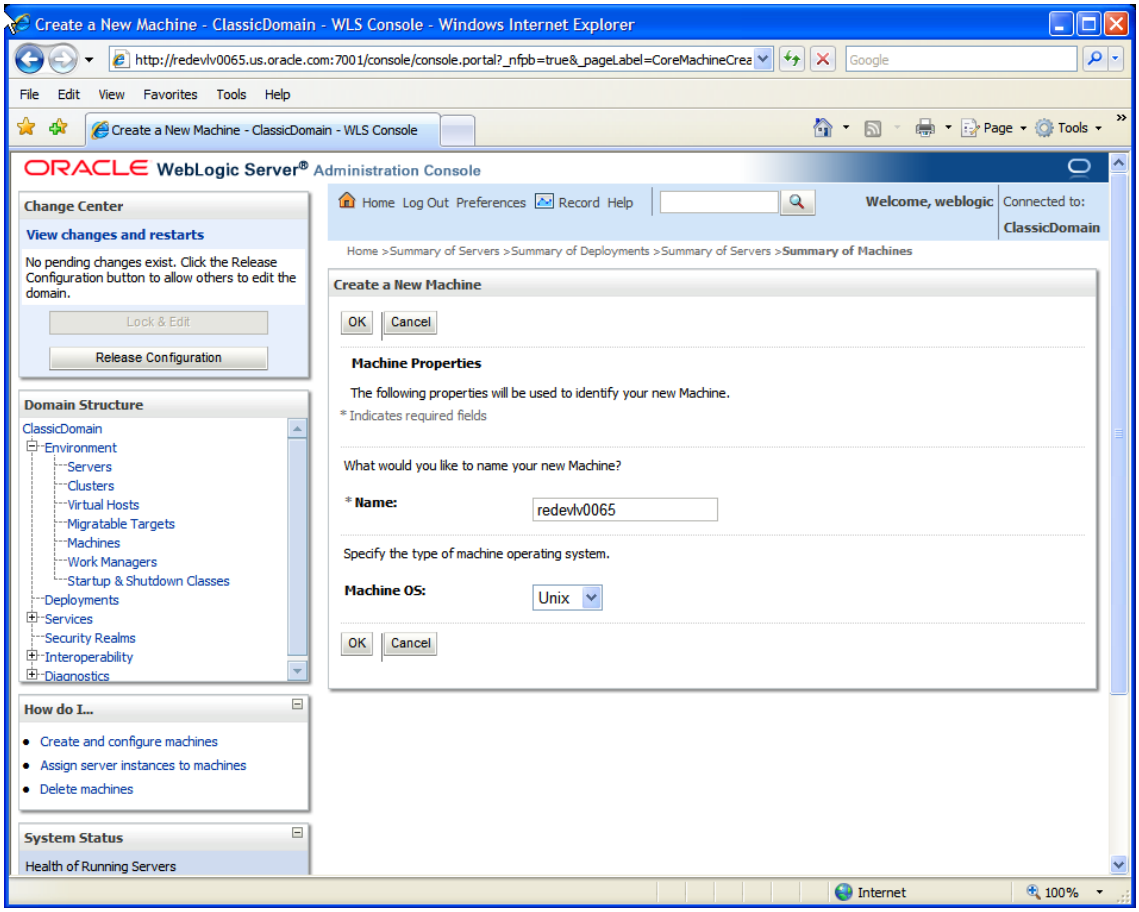


7. Click **Activate Changes** on the left side.

Install Node Manager

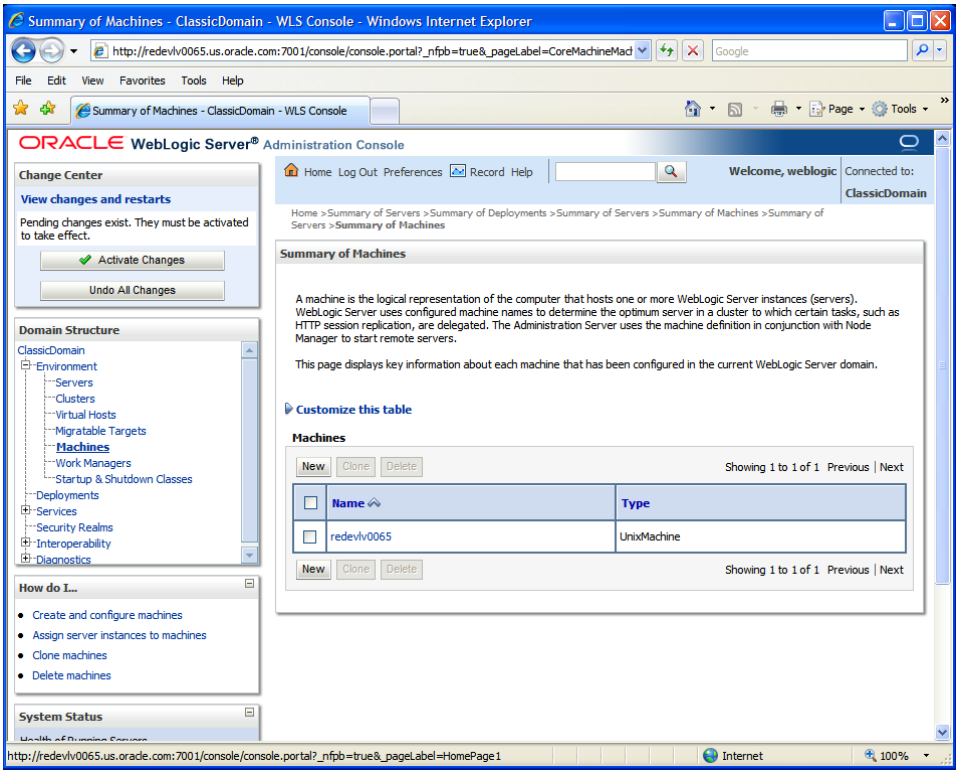
Install Node Manager if it was not created during domain install. The node manager is required so that the managed servers can be started and stopped through the admin console. Only one node manager is needed per WebLogic installation.

1. Log in to the admin console.
2. Click **Lock & Edit** button and navigate to Environments > Machines.
3. Click **New**.

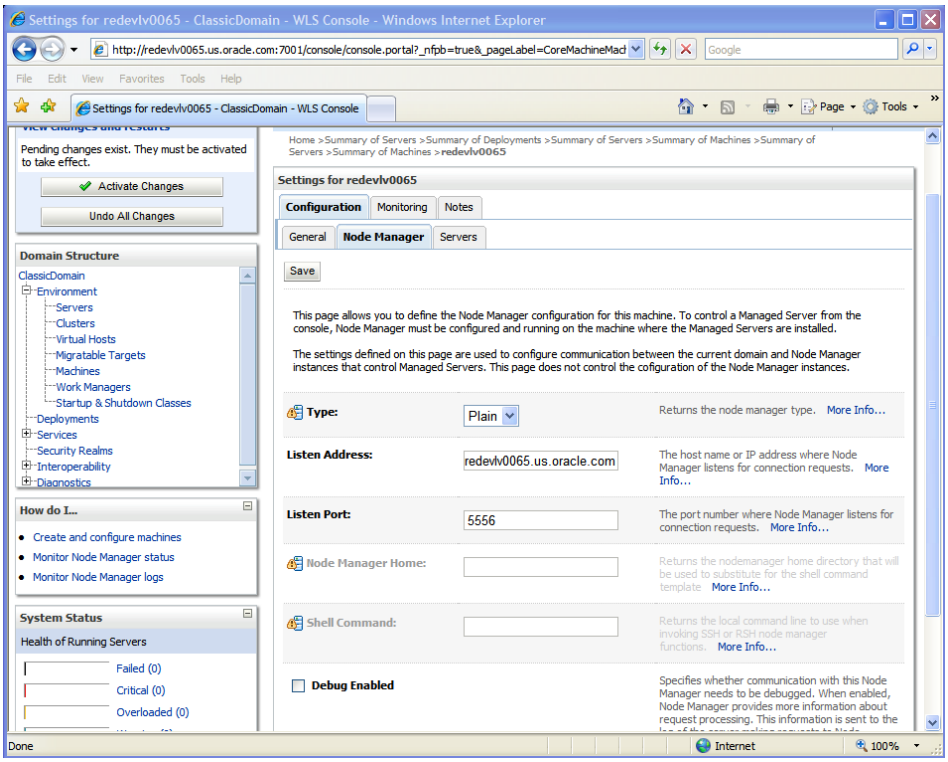


4. Set the following variables:
   - **Name**: Logical machine name
   - **Machine OS**: UNIX
5. Click **OK**.

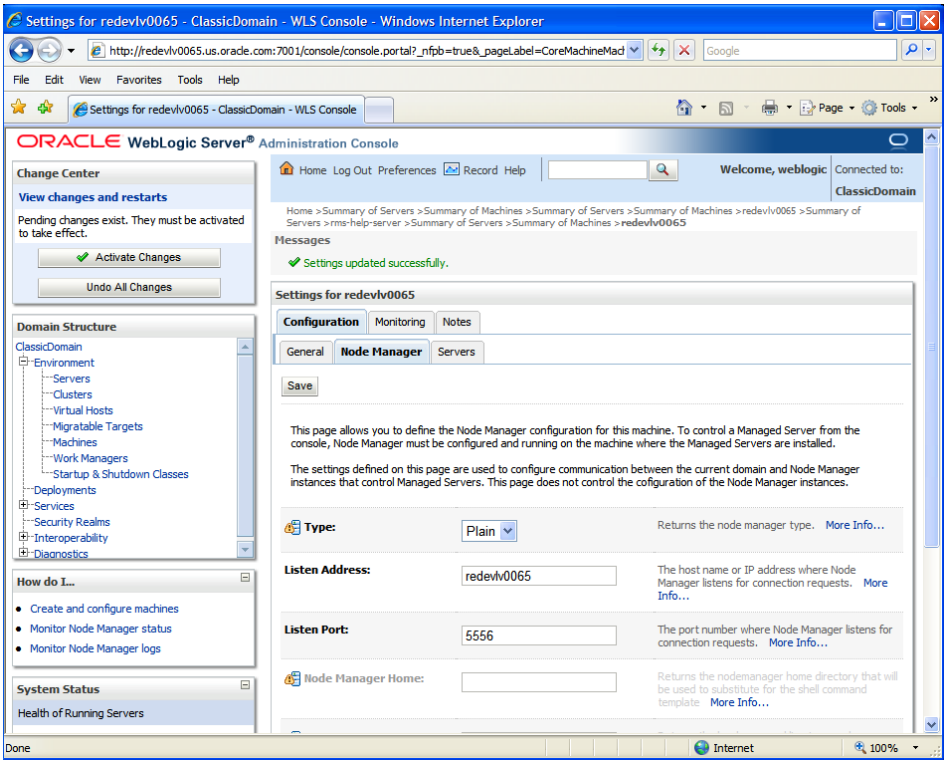
6. Click on the machine created.



7. Click the Node Manager tab and update the details below.
   - **Type:** Plain
   - **Listen Address:** Machine IP (ie: redevlv0065.us.oracle.com)
   - **Listen Port:** Node manager will be assigned a default port (for example, 5556)

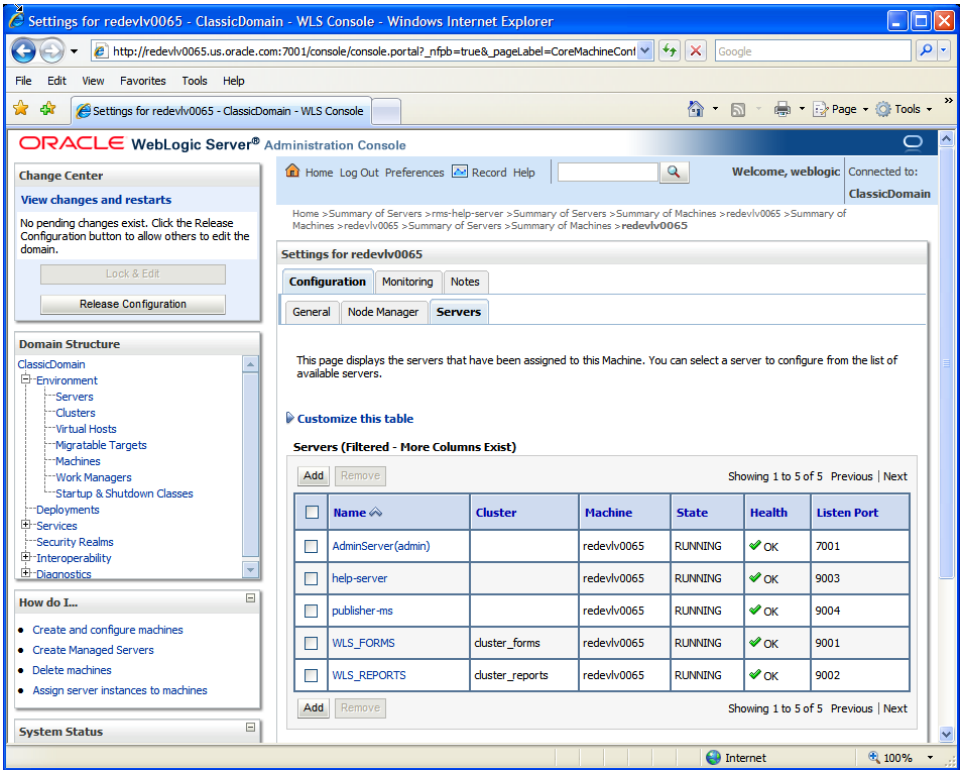


8. Click **Save**.



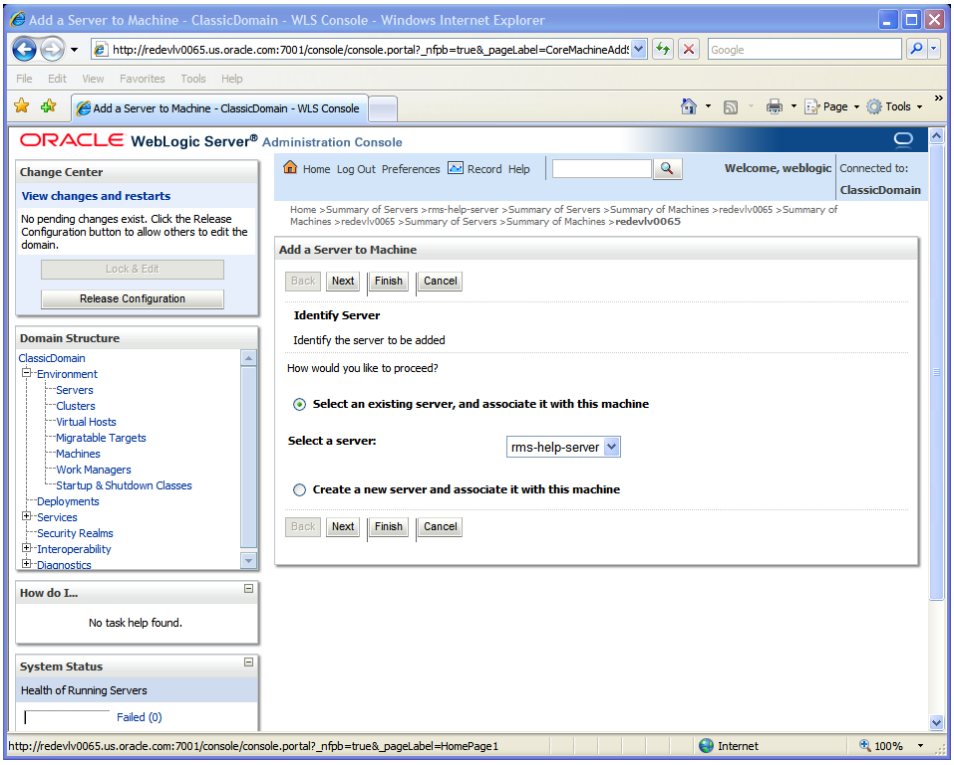
9. Click **Activate Changes**.
10. Click **Lock & Edit**.

11. Navigate to Environments->machines->click on the machine name and select the Servers tab.



12. Add the managed servers that need to be configured with the Nodemanager. Save the changes.

13. Click **Add**.



14. Set the following variables:
    - Server: name of server previously created (ie: rms-help-server)
15. Click **Next**. Click **Finish**.
16. Click **Activate Changes**.

> **Note:** In order to activate changes the server needs to be stopped if it is running:
>
> <WLS_HOME>/user_projects/domains/adf_domain/bin/stopManagedWebLogic.sh rms-help-server ${server_name}:${server_port}
>
> Go to each managed server that is being added to the machine and click the Server Start tab. In the Class Path box, add the following:
>
> `$CLASSPATH:<full-path-to-domain>/servers/<managed-server>`
>
> For example:
> `$CLASSPATH:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/ClassicDomain/servers/rms-help-server`

17. Click **Save**.
18. Click **Activate Changes**.

Create Staging Directory for RMS Application Server Files

To create the staging directory for the RMS application server files, complete the followig steps.

1. Log into the application server as a user with read and write access to the Weblogic files.
2. Create a staging directory for the RMS application installation software. There should be a minimum of 600 MB disk space available in this location.
3. Copy the file rms13application.zip from the RMS 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 rms13application.zip. This will create an rms/application subdirectory under STAGING_DIR.

Run the RMS Application Installer

Note: See Appendix: RMS Application Installer Screens for details on every screen and field in the application installer.

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/rms/application. This directory was created when the rms13application.zip file was expanded under STAGING_DIR.
3. Set and export the following environment variables.

Variable	Description	Example
DOMAIN_HOME	The location where Forms 11.1.1.3 domain has been installed.	DOMAIN_HOME= /u00/webadmin/product/10.3.3/WLS_Forms/user_projects/domains/ClassicDomain/ export DOMAIN_HOME
WLS_INSTANCE	The name of the managed server that contains Oracle Forms.	WLS_INSTANCE=WLS_FORMS Export WLS_INSTANCE
ORACLE_SID	The database/SID where the RMS schema resides.	ORACLE_SID=mydb
NLS_LANG	Locale setting for Oracle database client.	NLS_LANG=AMERICAN_AMERICA.UTF8 export NLS_LANG
JAVA_HOME	Location of a Java 6.0 (1.6.0) JDK.	JAVA_HOME= /u00/webadmin/java/jdk1.6.0_12 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 RMS 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 preinstall checks it will print out a summary containing these new environment variables:

> **Example:**
>
> MW_HOME=/u00/webadmin/product/10.3.3/WLS_Forms
>
> ORACLE_HOME=/u00/webadmin/product/10.3.3/WLS_Forms/as_1
>
> ORACLE_INSTANCE=/u00/webadmin/product/10.3.3/WLS_Forms/asinst_1
>
> DOMAIN_HOME=/u00/webadmin/product/10.3.3/WLS_Forms/user_projects/domains/ClassicDomain
>
> WLS_INSTANCE=WLS_FORMS
>
> ORACLE_SID=mydb
>
> JAVA_HOME=/u00/webadmin/java/jdk1.6.0_12

Verify that these environment variables are correct. If any of them are incorrect, you need to 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/10.3.3/WLS_Forms/user_projects/domains/ClassicDomain/bin/setDomainEnv.sh
>
> /u00/webadmin/product/10.3.3/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 RMS 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, you can check its log file: INSTALL_DIR/base/log/rms.app.install.<timestamp>.log. The INSTALL_DIR/base/error will contain information about possible failed compilations.
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`

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/10.3.3/WLS_Forms/user_projects/domains/ClassicDomain/config/
fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/config/forms/registry/ora
cle/forms/registry/Registry.dat
/u00/webadmin/product/10.3.3/WLS_Forms/user_projects/domains/ClassicDomain/config/
fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/config/formsweb.cfg

Have the WebLogic administrator stop WLS_FORMS and ohs1,
copy everything in /home/oretail/rms132/install/post
to /u00/webadmin/product/10.3.3/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/10.3.3/WLS_Forms
```

Resolving Errors Encountered During Application Installation

In the event a form or menu does not compile, go to INSTALL_DIR/base/error and see which objects didn't compile. To try and 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 box **Configure WebLogic** in the installer screens, you may need to clean up duplicate entries in the WebLogic formsweb.cfg file.

Clustered Installations – Post-Installation Steps

If you are installing the RMS 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 RMS application installer is referred to as *master node*. All other nodes are referred to as *remote nodes*.

To complete the RMS forms application install, the installer provided new versions of formsweb.cfg and the newly-created env files for the new RMS 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 RMS entries appended to this file by the installer. There is node-specific information in this file that is different between ORACLE_HOME installations.

Oracle Configuration Manager

The Oracle Retail OCM Installer packaged with this release installs the latest version of OCM.

The following document is available through My Oracle Support (formerly MetaLink). Access My Oracle Support at the following URL:

`https://support.oracle.com`

Oracle Configuration Manager Installer Guide (Doc ID: 1071030.1)

This guide describes the procedures and interface of the Oracle Retail Oracle Configuration Manager Installer that a retailer runs at the beginning of the installation process.

OCM Documentation Link

`http://www.oracle.com/technology/documentation/ocm.html`

RMS Reports Copied by the Application Installer

The application installer copies RMS report files to INSTALL_DIR/base/reports. These files should be installed into BI Publisher as documented in the RMS Reports chapter of this document.

Test the RMS Application

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

4

RMS Reports—Full

Prerequisites

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

- The RMS Database Schema installation into a running Oracle 11gR2 database platform must be completed.
- The RMS application components must be installed into a working WebLogic 10.3.3 domain.

Path References Used in this Section

The directory structures outlined in this section will vary according to your specific configuration of the target WebLogic domain. We will therefore use the labels outlined in the following table as references to directories used for the configuration of external dependencies for RMS 13.2 throughout the rest of this section.

> **Note**: The directory names for each of the labels outlined below for your installation environment.

Label	Description	Example Directory
<STAGING_DIR>	The application installation staging directory as defined for the source installation files	
<INSTALL_DIR>	The destination directory mentioned in the RMS Application installer for the RMS files.	
<BI_MEDIA>	Expanded source media for BI Publisher.	/tmp/BIPublisherSource/
<BI_DEPLOYMENT>	The deployment directory for BI Publisher as a WebLogic web application.	/u00/webadmin/RMS_BIP/
<WLS_JAVA_HOME>	The JDK location used by the WebLogic Server instance being installed to.	/u00/webadmin/java/jdk1.6.0_22.x64
<BI_REPOSITORY>	The BI Publisher reports repository. By default this is called XMLP.	/u00/webadmin/RMS_BIP/xmlpserver/XMLP
<WLS_SERVER_CONFIG_DIR>	The configuration base directory for the WebLogic server instance supporting the Oracle Forms environment.	/u00/webadmin/product/10.3.3/WLS_Forms/domains/ClassicDomain/config/fmwconfig/servers/WLS_FORMS
<FORMS_INSTALL_BASE>	The Oracle Forms base configuration directory.	<WLS_SERVER_CONFIG_DIR>/applications/formsapp_11.1.1/config

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 10.3.3. 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 OBIEE suite, please 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

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

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.

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

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

Extracting 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)

Directory	Component
<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.
   Example assuming that /u00/webadmin is the root of the installation:
   ```
   mkdir /u00/webadmin/RMS_BIP
   cd /u00/webadmin/RMS_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:
3. cp <BI_MEDIA>/manual/generic/xmlpserver.war <BI_DEPLOYMENT>
   For example,
   ```
   cp /tmp/BIPublisherSource/manual/xmlpserver.war /u00/webadmin/RMS_BIP
   ```

Creating an Exploded Directory for the Installation

You need to 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.
   Example assuming that /u00/webadmin is the root of the installation:
   ```
   cd /u00/webadmin/RMS_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/RMS_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 to update the catalog path in the xmlp-server-config.xml file, must be done before deployment. For more information on catalogs, refer to the BI Publisher Documentation.

3. Delete the war file.

   For example,

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

Configuring 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/RMS_BIP/xmlpserver
```

1. Assign appropriate permissions for the weblogic server instance user to have 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 the BI_HOME_DIR/xmlpserver/ 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/RMS_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 being used for the deployment (represented by WLS_JAVA_HOME in the example below). This is an optional task for users that plan to deploy their own customized reports for RMS using BI Publisher directly.
   For example,

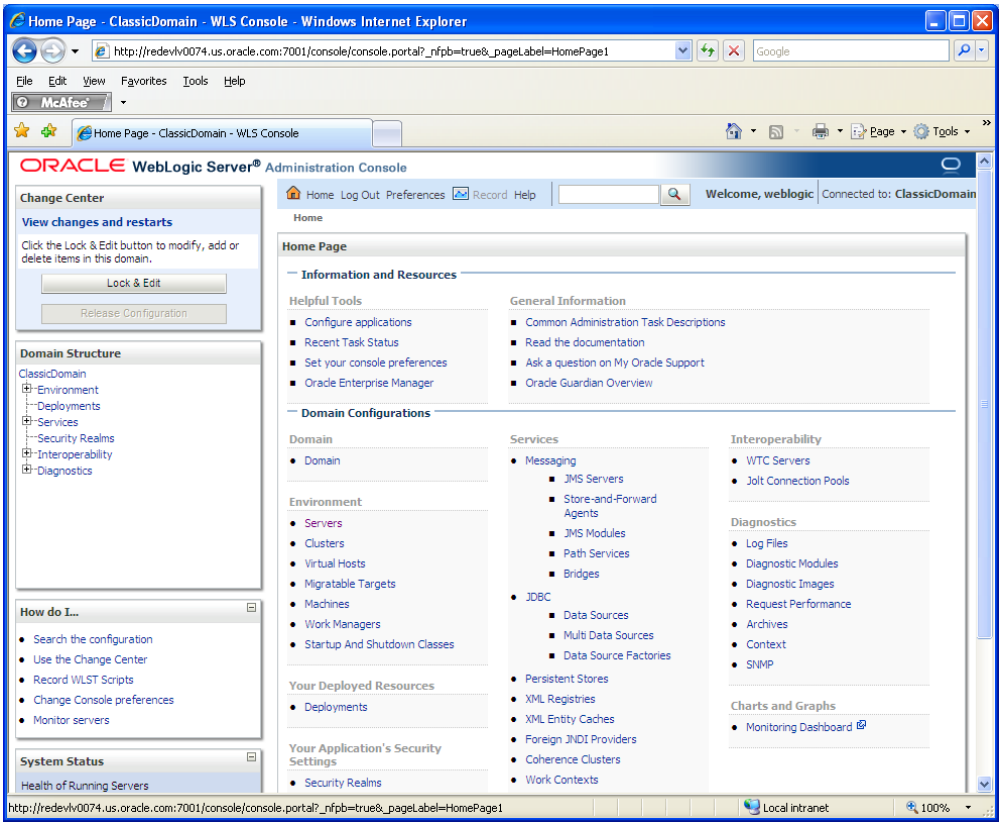
   ```
   cp - R /tmp/BIPublisherSource/fonts WLS_JAVA_HOME/jre/lib/fonts
   ```

> **Note:** This task will require a WLS server restart.

Install Managed Server in WebLogic

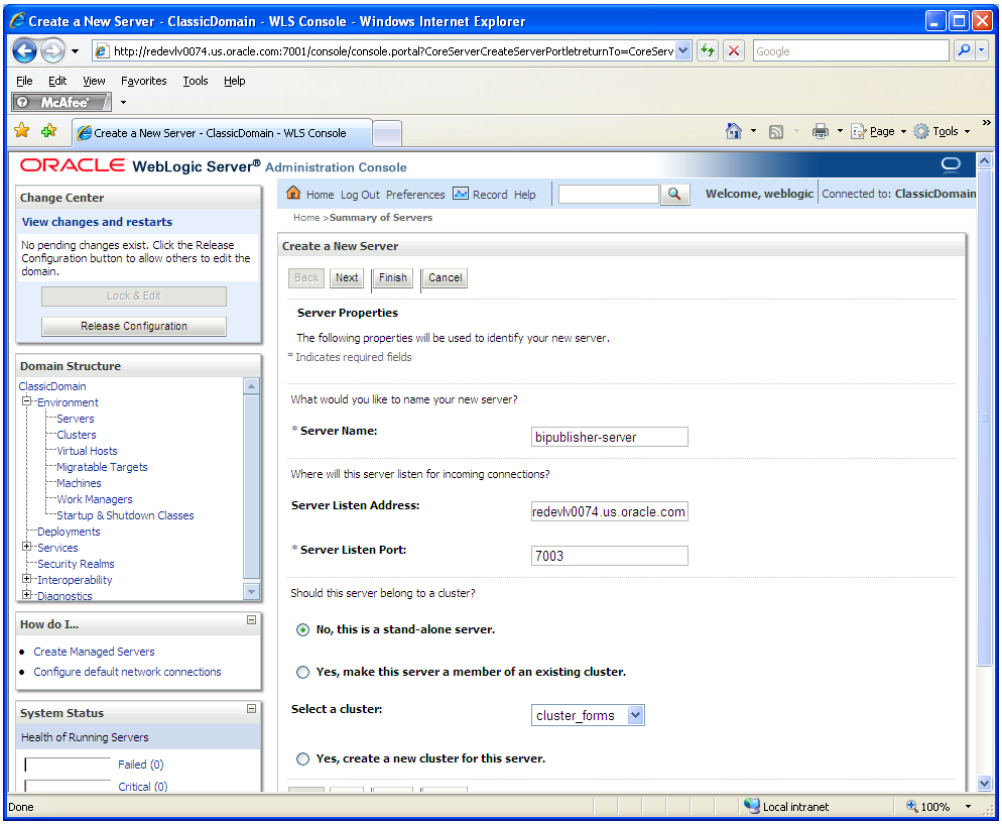
Before running the deployment of BI Application, you must install a managed server for deploying the BI application in WebLogic, if it was not created during the domain installation. Follow the steps below to install bipublisher managed server.

1. Log in to the admin console.



2. Click **Lock & Edit**.

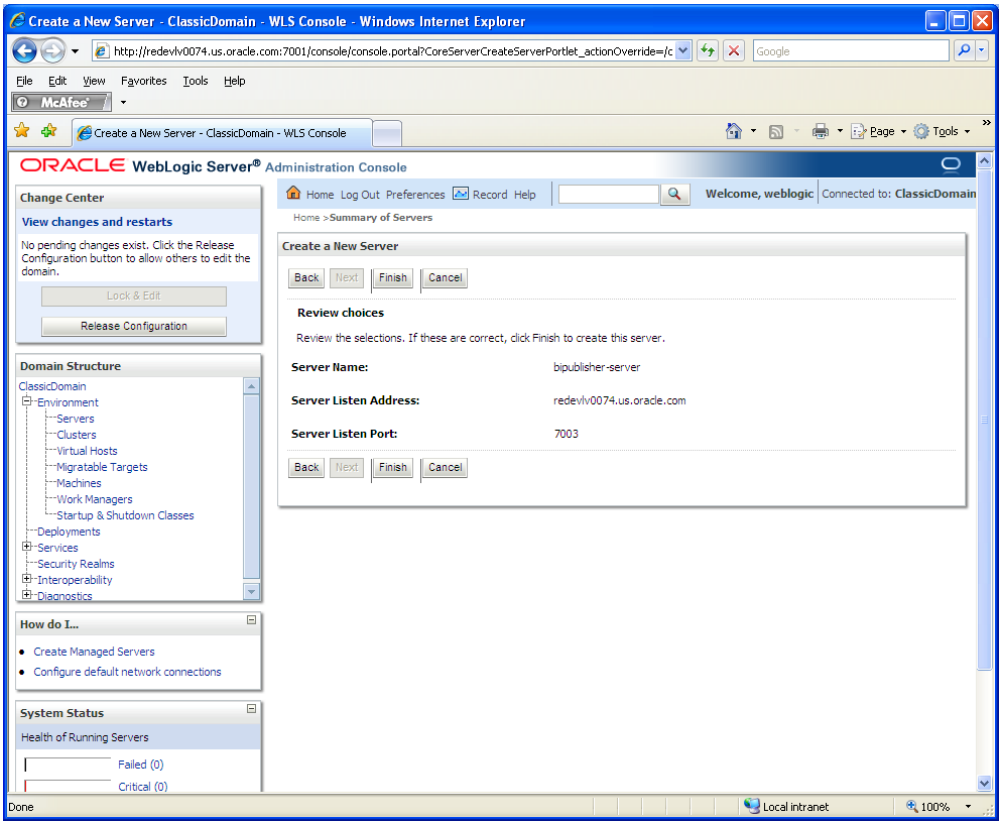
3. Navigate to Environment > Servers. Select new tab of the servers on the right side.



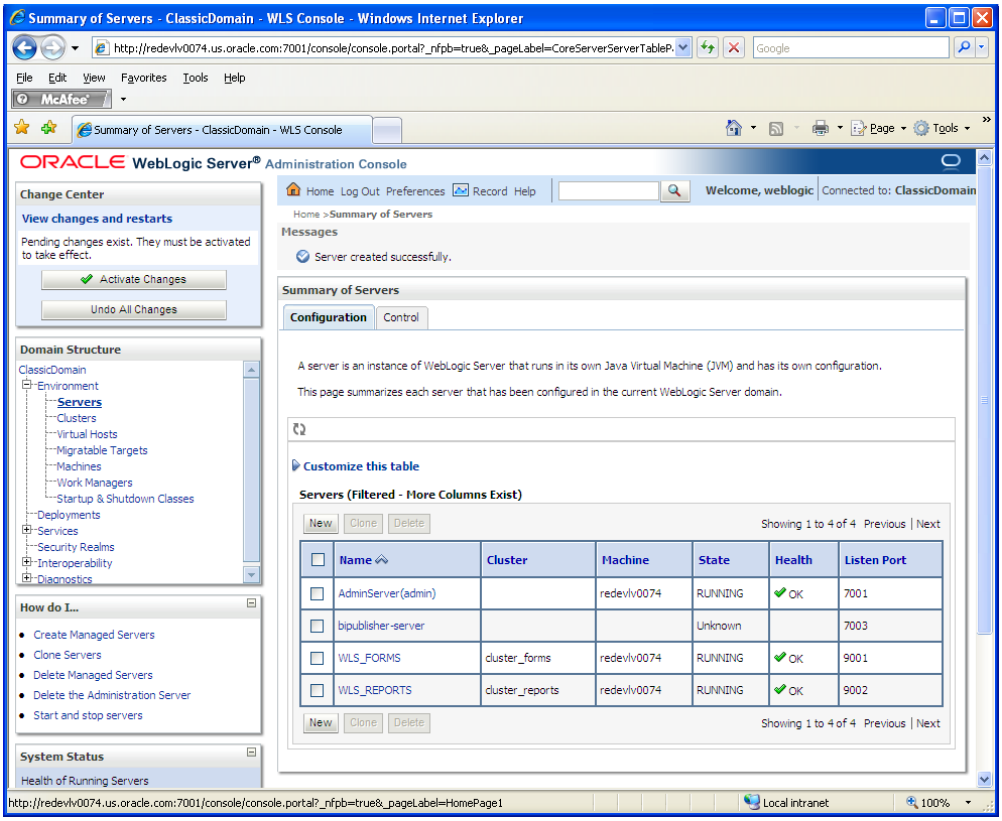
4. Set the following variables:
   - **Server Name**: This value should be specific to your targeted application (for example, bipublisher-server)
   - **Server Listen Address**: <weblogic server> (for example, redevlv0074.us.oracle.com)
   - **Server Listen Port**: A free port. Check for availability.

     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.)

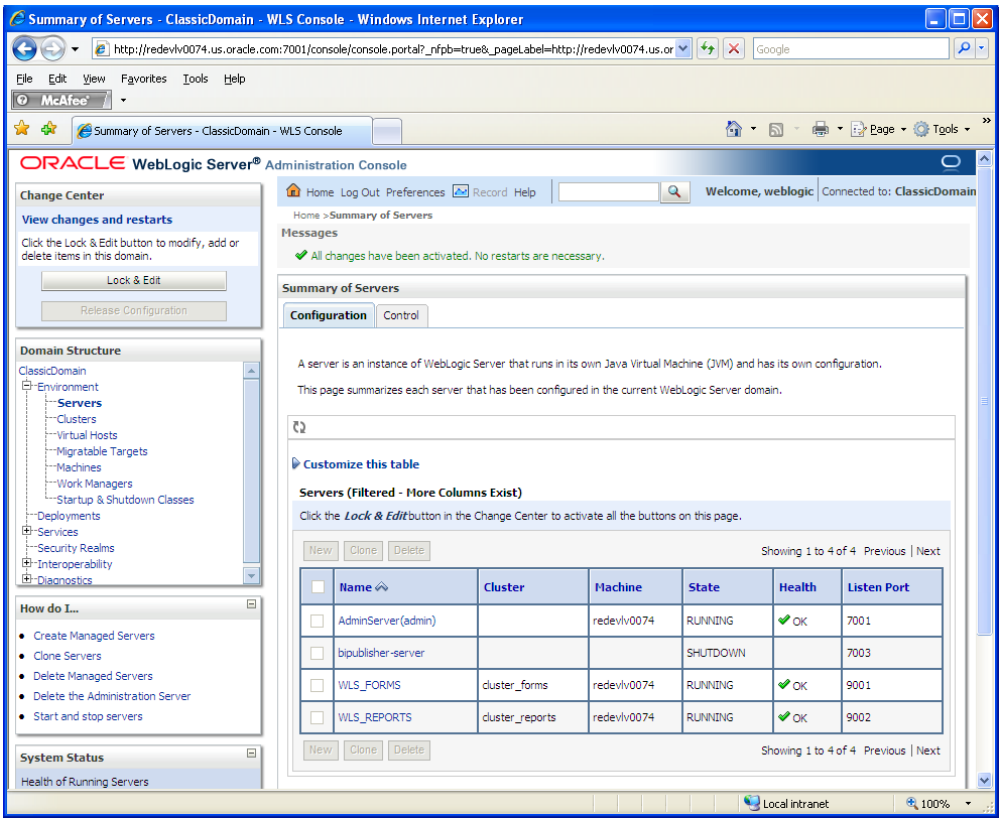
5. Click **Next**.


Create a New Server - ClassicDomain - WLS Console - Windows Internet Explorer
http://redevlv0074.us.oracle.com:7001/console/console.portal?CoreServerCreateServerPortlet_actionOverride=/c
File Edit View Favorites Tools Help
McAfee
Create a New Server - ClassicDomain - WLS Console
Page Tools
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
Health of Running Servers
Failed (0)
Critical (0)
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: ClassicDomain
Home >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: redevlv0074.us.oracle.com
Server Listen Port: 7003
Back Next Finish Cancel
Done
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.



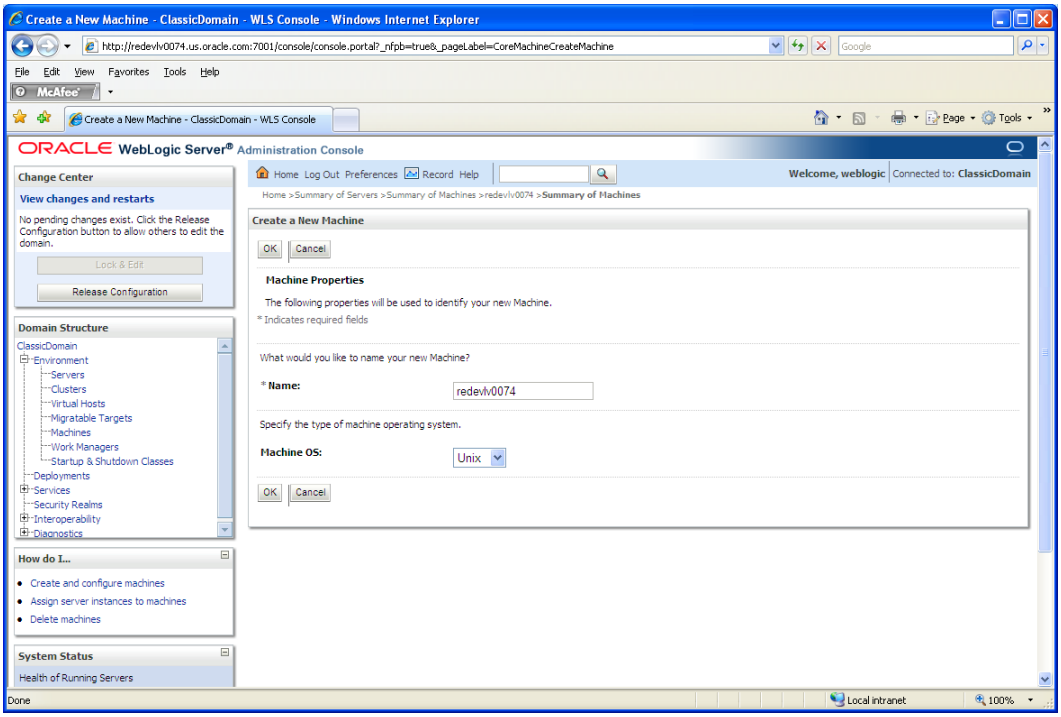
Install Node Manager

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

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

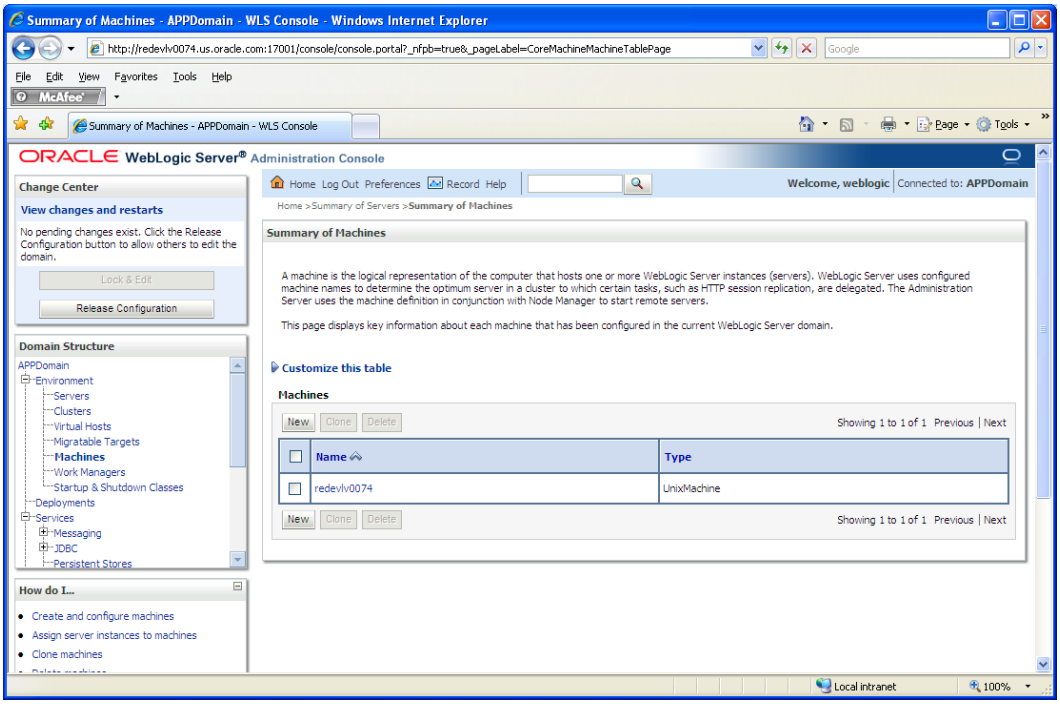
   The following page is displayed. Set the following variables:

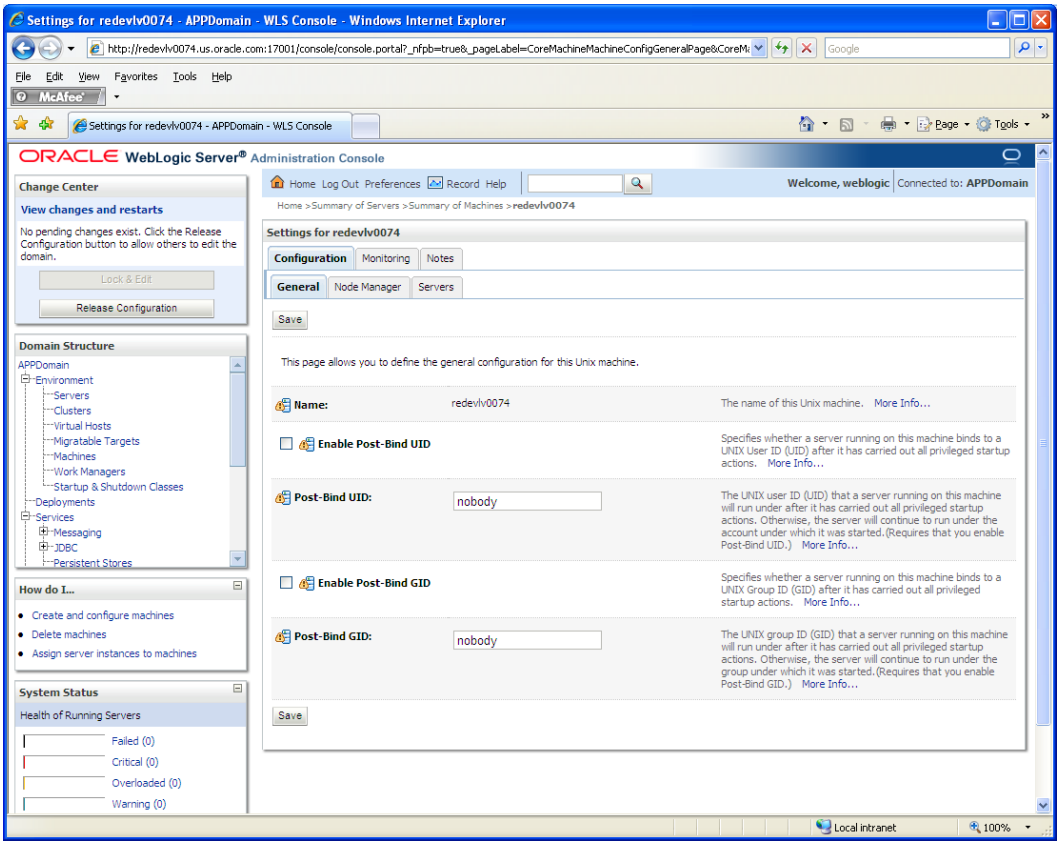
   - Name: Logical machine name
   - Machine OS: UNIX



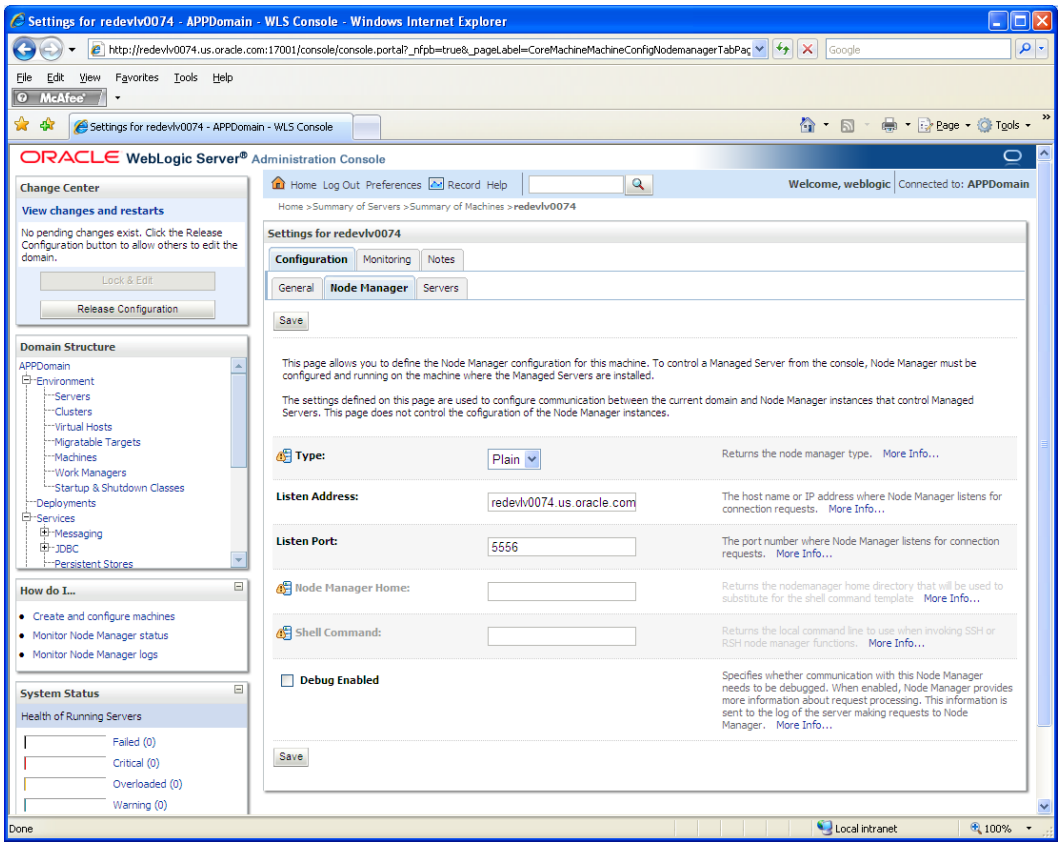
3. Click **OK** to activate the changes.

4. Click the machine created.



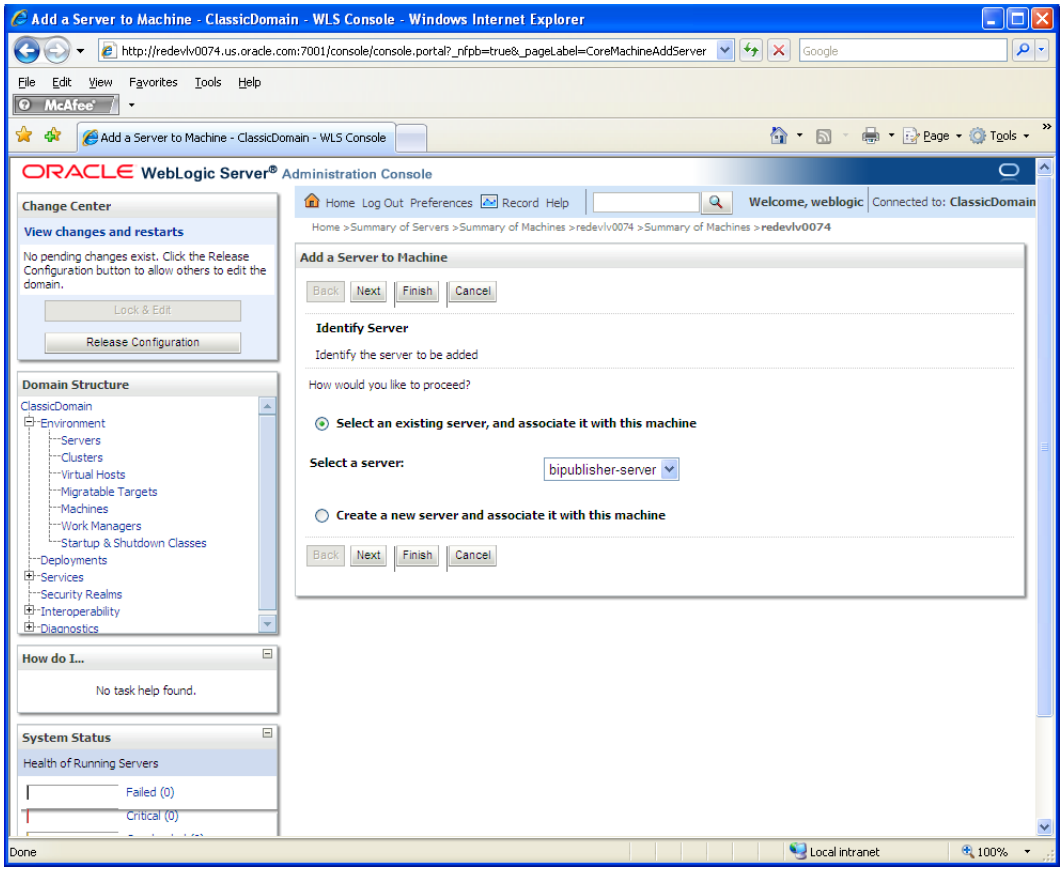


5. Click the Node Manager tab and update the details below.
   - **Type:** Plain
   - **Listen Address**: <weblogic server> (for example, redevlv0074.us.oracle.com)
   - **Listen Port**: Assign a port number. Default port is 5556.



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:

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

For example: `$CLASSPATH:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/<Domain_name>/servers/bipublisher-server`

---

13. Click **Save**.
14. Click **Activate Changes**.

Start the Managed Servers

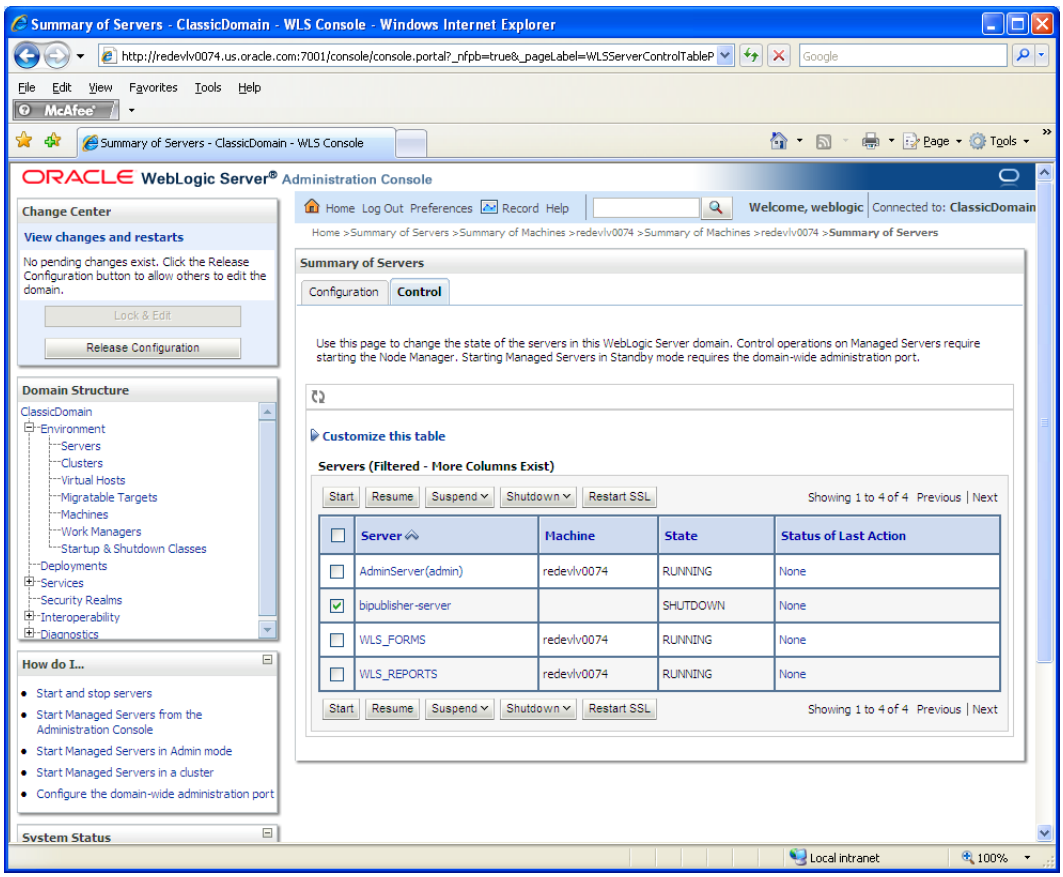
To start the managed servers, complete the following steps.

1. Start the Node Manager from the command line.

   ```
   $WEBLOGIC_HOME/wlserver_10.3/server/bin startNodeManager.sh
   ```

   After the Node Manager is started, the managed servers can be started through the admin console.

2. Navigate to Environments > Servers. Select bipublisher-server managed server. Click the Control tab.



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

Additional Set Up Steps Before Deploying the BI Application

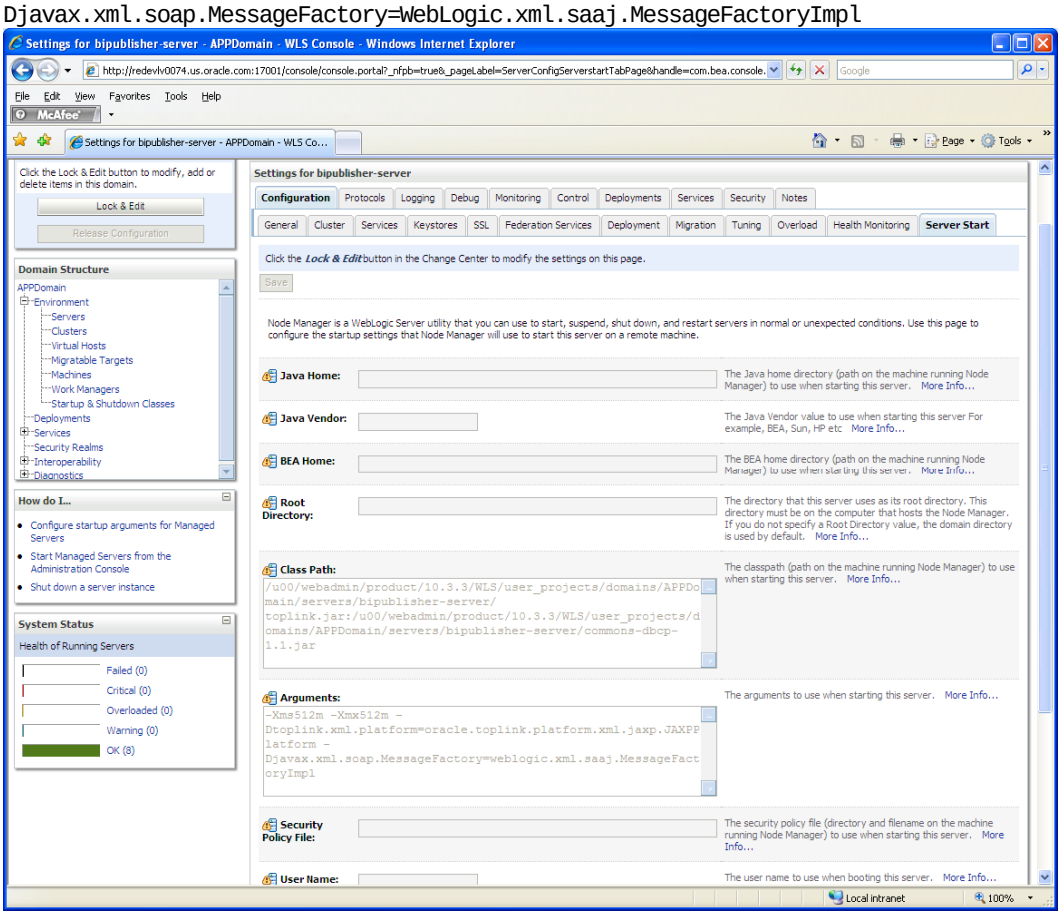
Following steps are the additional set up steps required before deploying the BI application in WebLogic.

1. Shutdown 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.

   ```
   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 admin console is available in the path: Weblogic Administration console->Servers->BIPublisher managed server->Server Start-> Arguments)

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



5. Restart the WebLogic server.

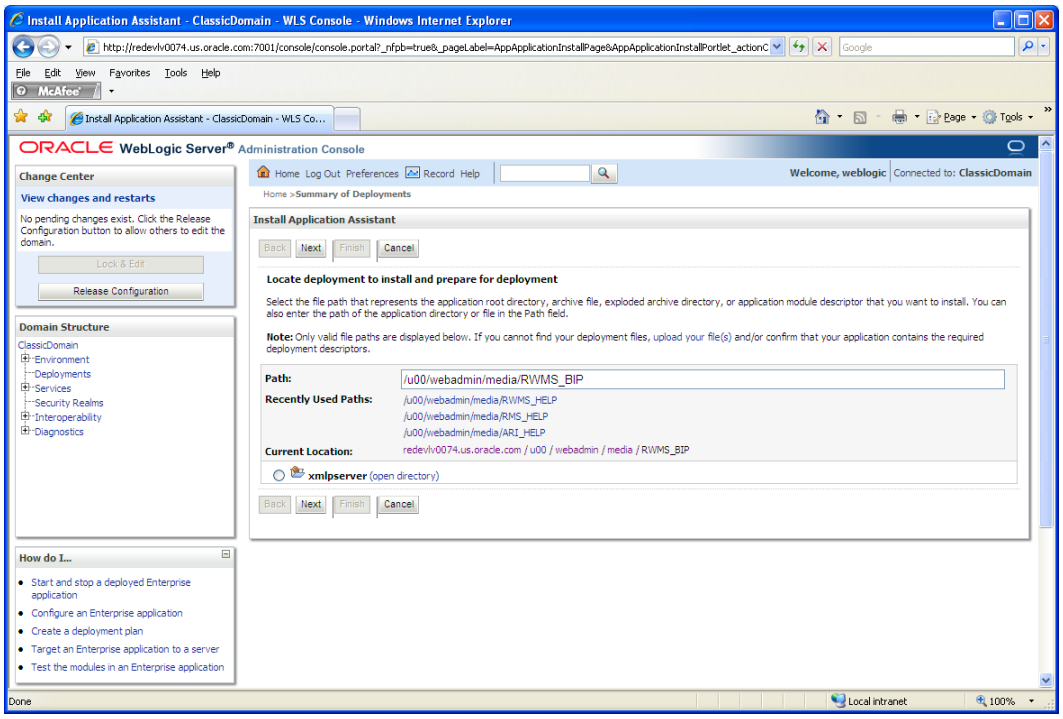
Deploying the BI Application in WebLogic

The exploded archive directory created above in "Creating an Exploded Directory for the Installation" must now be deployed into the bipublisher managed server of WebLogic. Deployment can be achieved in a number of ways but we will use the WebLogic Administration Console and the following steps.

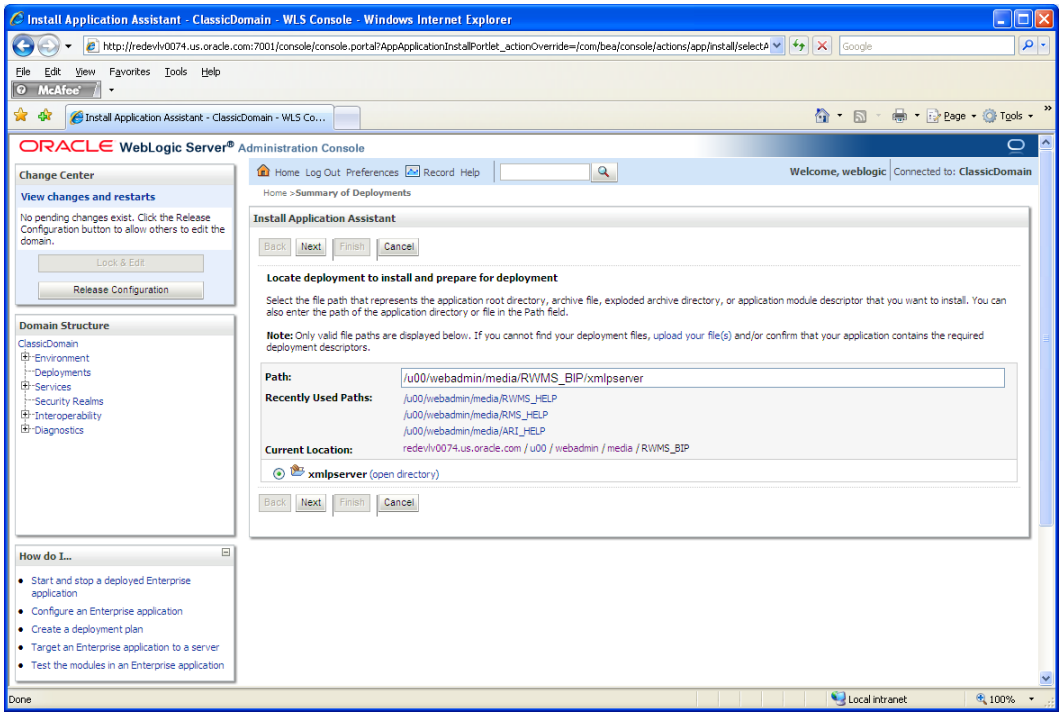
Open the WebLogic Administration console web page by typing the appropriate URL for the WebLogic admin server.

For example,
`http://wls_srv:7001/console`

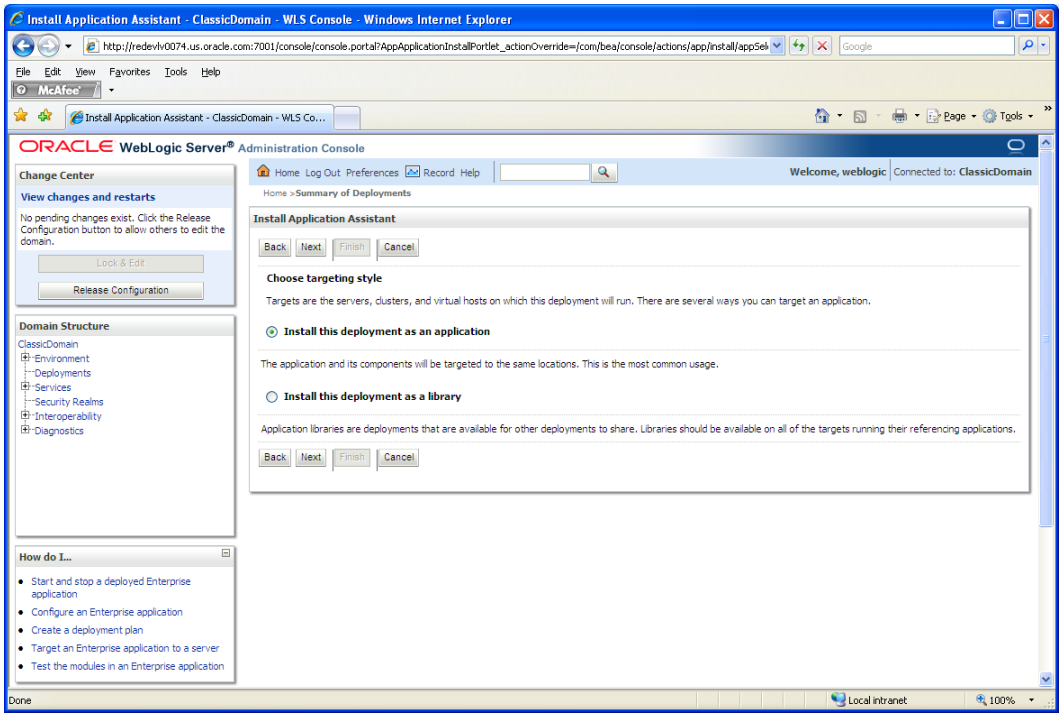
1. Log on to the console using an administrator user name and password.
2. In the Change Center of the Administration Console, click **Lock & Edit**.
3. In the left pane of the Administration Console, click **Deployments**.
4. In the right pane, click **Install**. The following screen should be displayed.



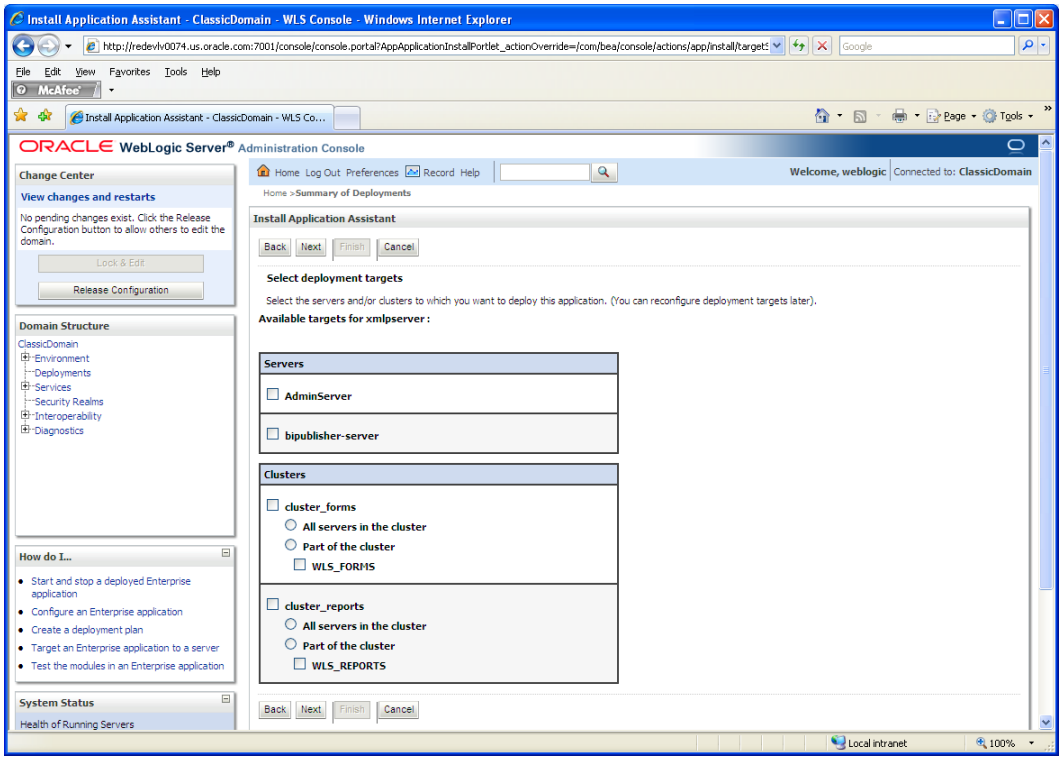
5. Select the exploded directory referenced by <BI_DEPLOYMENT>/xmlpserver. Click **Next**.



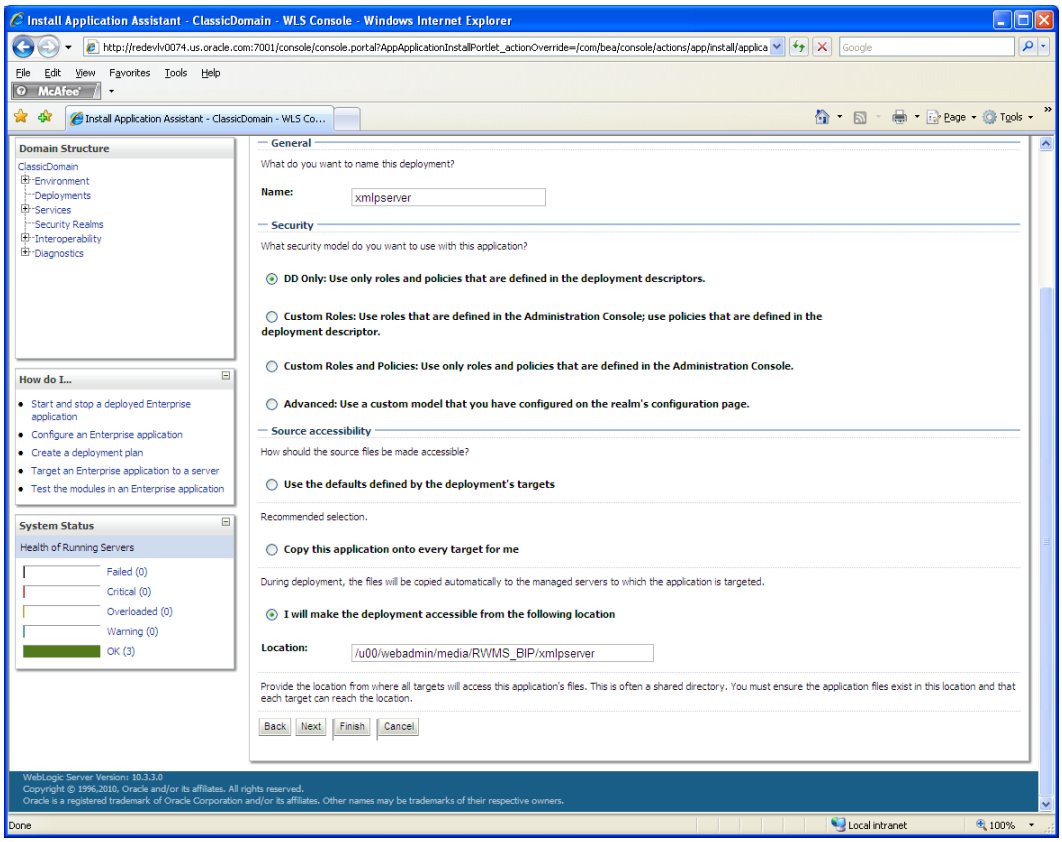
6. Select the Install this deployment as an application option. Click **Next**.



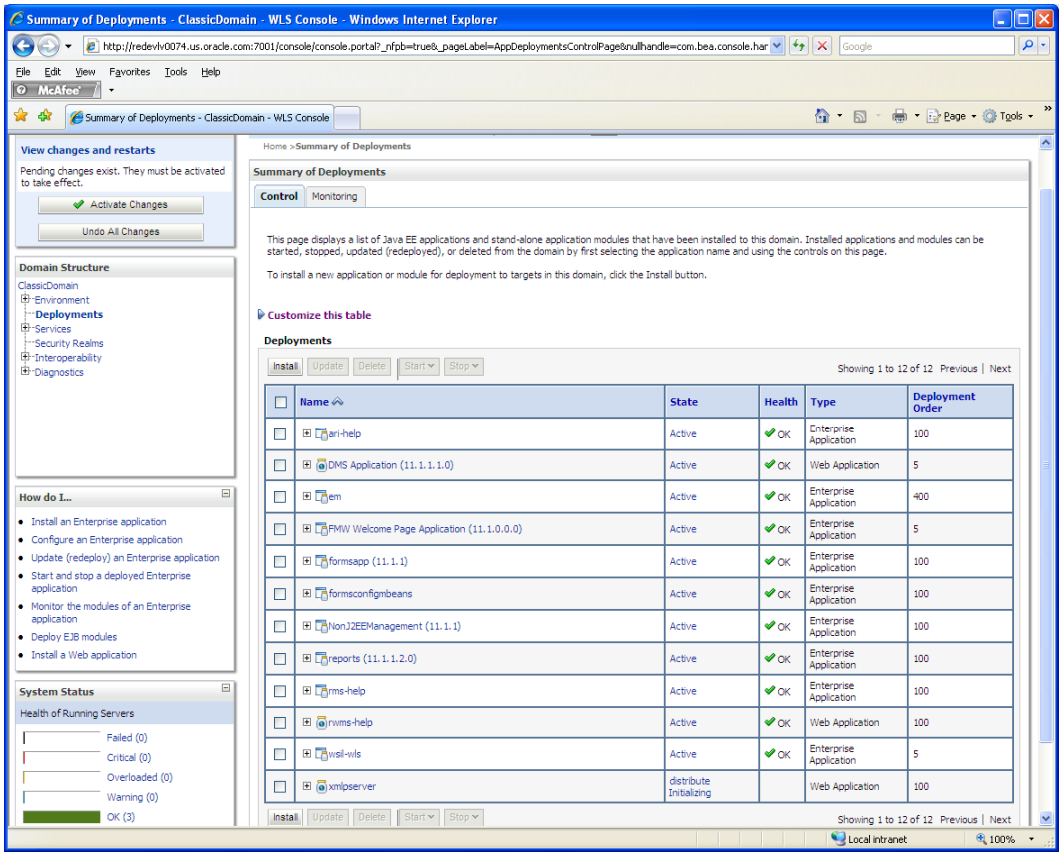
7. Select the bipublisher managed server as the deployment target. Click N**ext**.



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

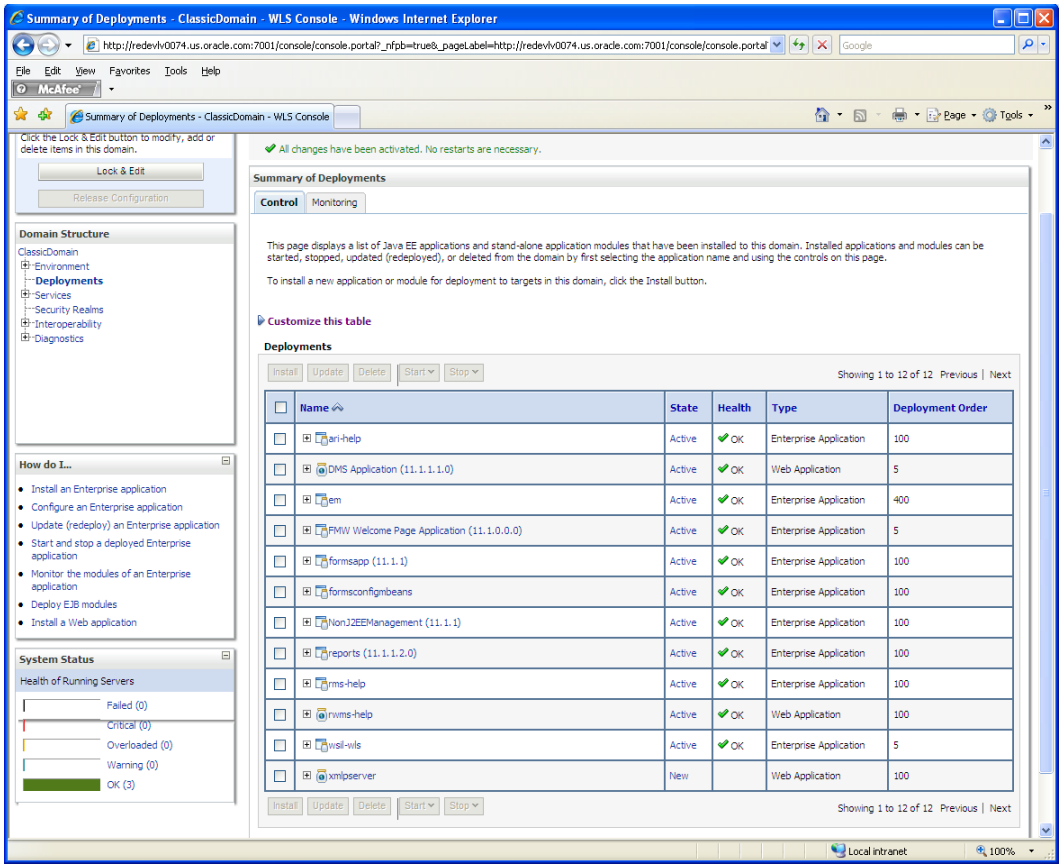


9. Click **Finish**.

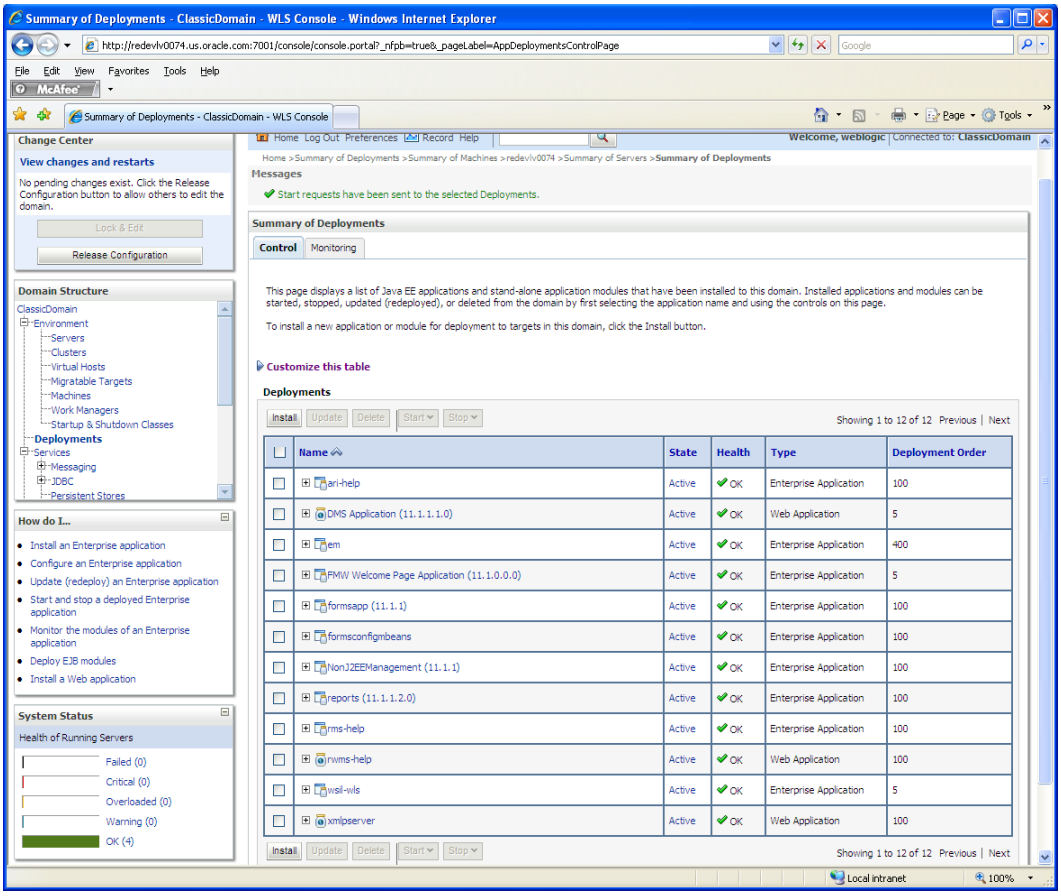


10. To activate these changes, in the **Change Center** of the Administration Console, click **Activate Changes**.

11. On the left side of the console, select Deployments. A list of deployments should now be showing in the table on the right.



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



13. Launch BI Publisher using the appropriate URL for the WebLogic server appended by the web application context "/xmlpserver".

    For example:

    `http://wls_srv:7003/xmlpserver`

Installing the RMS BI Publisher Templates

In this section we will outline how the RMS report templates are installed into the appropriate BI server repositories.

Report files are placed by the application installer in the directory - " `INSTALL_DIR/base/reports` " and have to be copied into the newly created directory.

1. Create a RMS directory to hold the reports under <BI_REPOSITORY>/Reports/Guest.

   Example <BI_REPOSITORY>Reports/Guest/RMS13

2. Change directory to the INSTALL_DIR/base/reports used for the application install. This directory contains subdirectories whose names reflect the names of report templates provided with RMS.

3. Copy each report directory into the directory created above

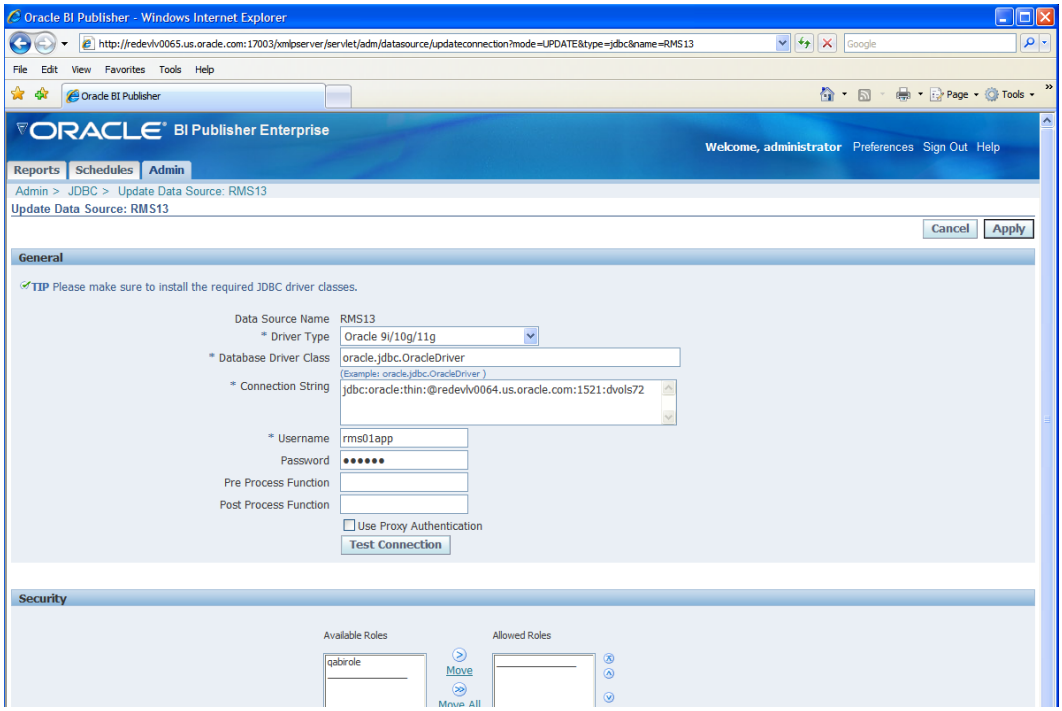
   For example,

   ```
   cp -R *  /u00/webadmin/RMS_BIP/xmlpserver/XMLP/Reports/Guest/RMS13/
   ```

Configuring the RMS JDBC connection

Follow the below steps to configure JDBC connection for RMS Data Source name. This datasource RMS will be used for RMS 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 will be displayed:



3. Enter the appropriate details for the RMS data source. Once the data is entered, click Test Connection to test the connection.

Configuring the BIPublisher Scheduler

Complete the following tasks for scheduler configuration:

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 following screen. Enter the appropriate database connection details and test the connection as previously done. If this connection operates successfully, save the connection details and proceed by clicking the Install Schema button. This installs the schema for BI publisher using the RWMS connection details.

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

3. Click the **Admin** tab. Click **Report Repository** under System Maintenance. The Path variable should be set as part of the BI Publisher install, REPORTS_DIR.
4. Add the following values to the <installation name>.env file located here: $WLS_HOME/user_projects/domains/<domain name>/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_11.1.1/config/<installation name>/<installation name>.env
   - ORACLE_RMS_REPORTS_HOST=http://<server>:<port>/

     For example, ORACLE_RMS_REPORTS_HOST=http://redevlv0065.us.oracle.com:9004/
   - ORACLE_RMS_RWSERVER=/<location to RMS directory>/

     For example, ORACLE_RMS_RWSERVER=xmlpserver/Guest/RMS13/

Part II: Upgrade Installation

The database portion of RMS can be upgraded from release 13.1.2 to release 13.2. Part II of this guide details the steps needed to perform an upgrade installation of RMS. For additional information on the upgrade, see the *Oracle Retail Upgrade Guide* (Doc ID 1073414.1) at My Oracle Support (formerly MetaLink).

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.

Part II contains the following chapters:

- Chapter 7, RMS Database Installation – Upgrade
- Chapter 8, Batch Installation Tasks – Upgrade
- Chapter 9, Application Server Installation Tasks – Upgrade
- Chapter 10, RMS Reports Installation – Upgrade
- Chapter 11, Data Migration

For information about a full installation, see Part I.

1

RMS Database Installation—Upgrade

There are two different methods to use for installing the RMS 13.2 database schema upgrade. Option 1 uses the installer to apply the upgrade. Option 2 uses the upgrade controller scripts directly.

> **Note:** The patching mechanism has been updated for the 13.2 release. Any patches that were released prior to 13.2 (For example, 13.1.1 and 13.1.2) will not be compatible with this installer.

Preparing to upgrade the RMS Database

RMS 13.2 utilizes Oracle Advanced Queuing, and so requires additional privileges for RMS owning schemas and generic users. Prior to running the upgrade, ensure that the following privileges have been granted:

- RMS Owning Schemas
  - grant AQ_ADMINISTRATOR_ROLE to <user>;
  - grant execute on dbms_aq to <user>;
  - grant execute on dbms_aqadm to <user>;
- RMS Generic Schemas
  - grant enqueue any queue to <user>;
  - grant manage any queue to <user>;
  - grant dequeue any queue to <user>;
  - grant execute on dbms_aq to <user>;
  - grant execute on dbms_aqadm to <user>;

Option 1: Upgrade RMS Database using the Installer

The RMS 13.2 database schema installer may be used to apply the RMS upgrade. The installer should only be used to apply the upgrade if the schema being upgraded does not contain customizations or hotfixes. The upgrade may also be applied outside of the installer by calling the controller scripts directly. *See* Option 2: Upgrade RMS Database using Controller Scripts later in this chapter for details on this method.

In this section, STAGING_DIR refers to the location where the RMS 13.2 database schema installer is expanded.

Before you apply the RMS 13.2 upgrade:

- Make a backup of all your objects and database schema.
- Check that RMS 13.1.2 is installed.
- Review the enclosed RMS 13.2 Upgrade Release Notes (rms-132-rn.pdf).
- Review each of the enclosed defect documents.
- Make sure any applications that connect to the RMS schema are shut down. This includes RPM, ReIM, Allocation, RIB, and anything else that could be using the schema.

Before copying over any files:

- Note whether customizations have been made to the module. If so, then the customizations must be reapplied over the new version of the module (or the fix may need to be applied to the custom version of the code).
- Copy the original files to a different directory before copying over them in case they need to be referred to at a later date.

Create Staging Directory for RMS Database Schema Files

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

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the MOM 13.2 Upgrade. There should be a minimum of 1500 MB disk space available in this location.
3. Copy the rms13dbschema.zip file from the RMS 13.2 release to the staging directory. This is referred to as STAGING_DIR when installing database software.
4. Change directories to STAGING_DIR and extract the rms13dbschema.zip file. This creates an rms/dbschema subdirectory under STAGING_DIR.

Run the RMS Database Schema Installer

To run the RMS database schema installer, complete the following steps.

> **Note:** See Appendix: RMS Batch Installer Screens for details on screens and fields in the RMS database schema installer.

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

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

Verify the ORACLE_HOME and ORACLE_SID variables after running this script.

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

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

Variable	Description	Example
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. To run the installer in GUI mode using an X server, the XTEST extension must be enabled. This setting is not always enabled by default in your X server. See Appendix: Common Installation Errors for more details.
5. If the installer has already been run in this location you may wish to back up the ant.install.properties file. The settings from the RMS 13.2 full install might be in this file, and running the installer again for the upgrade clears out some of the settings that are not used by the installer's patch mode.
6. Run the install.sh script to start the installer.

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

   install.sh [text | silent]

   If prompted about resuming previous installation, respond with no.

Select the Patch option on the Full Install or Patch Option screen.

7. On the Apply a Patch page for each product, provide the path to STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2 /<product>. This directory should contain a <product>_controller.ksh file (for example, rms_controller.ksh), which the installer runs to apply the RMS 13.2 upgrade.
8. After the installer is complete, you can check its log file: rms-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`

Resolving Errors Encountered During Database Schema Installation

Errors encountered during upgrade installation should be resolved outside of the installer using SQL*Plus. The installer's resume function is only useful for full installs.

Option 2: Upgrade RMS Database using Controller Scripts

While the installer can be used to apply the entire RMS database upgrade, there are situations in which it is better to use run the upgrade directly with the scripts released in the upgrade. The installer calls start-all ksh scripts named <product>_controller.ksh which run all of the files in the upgrade. If there are any customizations or hotfixes in the schema then certain statements in the upgrade may result in errors. In this situation it is better to investigate where the conflicts are and fix the SQL scripts accordingly.

Before you apply the RMS 13.2 upgrade:

- Make a backup of all your objects and database schema.
- Check that RMS 13.1.2 is installed.
- Review the enclosed RMS 13.2 Upgrade Release Notes (rms-132-rn.pdf).
- Review each of the enclosed defect documents.

Before copying over any files:

- Note whether customizations have been made to the module. If so, then the customizations must be reapplied over the new version of the module (or the fix may need to be applied to the custom version of the code).
- Copy the original files to a different directory before copying over them in case they need to be referred to at a later date.

Create Staging Directory for RMS Database Schema Files

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

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the MOM 13.2 Upgrade. There should be a minimum of 1500 MB disk space available in this location.
3. Copy the rms13dbschema.zip file from the RMS 13.2 release to the staging directory. This is referred to as STAGING_DIR when installing database software.
4. Change directories to STAGING_DIR and extract the rms13dbschema.zip file. This creates an rms/dbschema subdirectory under STAGING_DIR.

Run the RMS Database Controller Scripts

To run the RMS database controller scripts, complete the following steps.

1. Change directories to STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2.
2. Source the oraenv script to set up the Oracle environment variables (ORACLE_HOME, ORACLE_SID, PATH, etc)

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

3. 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

4. Set and export the NLS_LANG environment variable.

   Example: NLS_LANG=AMERICAN_AMERICA.UTF8
   export NLS_LANG

5. For each product you want to upgrade, configure the individual controller.cfg files. To do this, copy STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>/templates/controller.cfg to STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>/controller.cfg
6. Open the controller.cfg file you just created and replace the tokens for the following variables with the appropriate values:
    a. Export PATCH_DIR= STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>
    b. export SCHEMA_OWNER=<The name of the RMS schema>
    c. export MMUSER=<The name of the schema to Upgrade >. For RMS, RPM, ReIM, and Alloc_RMS, this will be the RMS schema. For Alloc, this will be the Allocation schema
    d. export ORACLE_SID=<SID for the database the MMUSER schema resides in>
    e. export TNS_ADMIN=/path/to/wallet/files/dir/
    f. export UP=/@<Schema Owner Wallet Alias>

    Note: See the document, "oracle_wallet_setup_for_mom.doc," for instructions to set up the database wallet.
7. The product upgrades should be run in the following order: RMS, RPM, ReIM, Alloc_RMS, Allocation. While you can choose not to run any of these upgrades, all of the patches other than RMS depend on the RMS 13.2 Upgrade being run. If you upgrade RMS you should also upgrade RPM; there is also a dependency between Alloc_RMS/Alloc. For each product you wish to upgrade, cd to STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product> and run the following commands:

    ```
    For RMS run: $ ./rms_controller.ksh DBO N
    For RPM run: $ ./rpm_controller.ksh DBO Y
    For ReIM run: $ ./reim_controller.ksh DBO Y
    For Alloc_rms run: $ ./alloc_controller.ksh DBO Y
    For Allocation run: $ ./alloc_rms_controller.ksh DBO Y
    ```

    Note: The controllers should be run in this order.
8. If the installation fails for any of the upgrades before completion, look at the logs in the STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>/error and STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>/log directories to determine the source of the error. You can continue the upgrade by rerunning the <product>_controller.ksh file, but only if the files generated in the STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>/processed directory from the last upgrade attempt are still there. Any scripts that ran previously will be skipped. If you wish to start a new upgrade, delete all files in the STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/<product>/processed directory.

2

Batch Installation Tasks—Upgrade

There are two different methods to use for installing the RMS 13.2 Batch Upgrade. Option 1 uses the installer to apply upgrade. Option 2 compiles the batch directly.

Option 1: Use Batch Installer to Upgrade

The installer should only be used to apply the upgrade if the batch being upgraded does not contain customizations or hotfixes. If the upgrade is applied to customizations, they will be overwritten.

Before you apply the RMS 13.2 Batch upgrade:

- Make a backup of all your Batch files.
- Review the enclosed RMS 13.2 Upgrade Release Notes (rms-132-rn.pdf).
- Review each of the enclosed defect documents.

Before running the installer:

- Note whether customizations have been made to the module. If so, then the customizations must be reapplied over the new version of the file (or the fix may need to be applied to the custom version of the code).
- Copy the original files to a different directory before installing in case they need to be referred to at a later date.

Run Batch Installer

The steps to run the Batch installer for an upgrade are the same as the steps for a full install. See the Batch Installation Tasks—Full chapter for details.

Option 2: Compile RMS Batch Directly

Note: Warning messages may appear during the compilation of the batch. These warnings can be ignored if the batch executables are successfully generated.

Create Staging Directory for RMS Batch Upgrade Files

To create a staging directory for RMS batch upgrade files, complete the following steps.

1. Log into the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS 13.2 Batch Upgrade. There should be a minimum of 14 MB disk space available in this location.
3. Copy the rms1320batchpatch.zip file from the RMS 13.2 release to the staging directory. This is referred to as BATCH_PATCH_DIR when upgrading a database schema.
4. Change directories to BATCH_PATCH_DIR and extract the rms1320batchpatch.zip file.

Set Environment Variables

> **Note:** INSTALL_DIR is the location where RMS 13 batch was installed.

Make sure the following variables are set. The RMS 13.2 batch installer should have created a batch.profile file located at INSTALL_DIR/batch.profile. This profile script can be used to set all of the environment variables listed below.

> **Example**: cd <INSTALL_DIR>
>
> . ./batch.profile

Variables set by batch.profile:

- PATH must include make, makedepend and the C compiler
- MMHOME=INSTALL_DIR
- MMUSER=RMS Schema Owner
- ORACLE_HOME=Location of Oracle DB install
- ORACLE_SID=The Oracle Sid for the RMS database
- UP=/@< Schema Owner Wallet Alias >
- TNS_ADMIN=/path/to/wallet/files/dir/

AIX:

- LIBPATH=$ORACLE_HOME/lib:$MMHOME/oracle/lib/bin:$LDPATH
- OBJECT_MODE=64
- LINK_CNTRL=L_PTHREADS_D7

Linux:

- LD_LIBRARY_PATH=$ORACLE_HOME/lib: $MMHOME/oracle/lib/bin:$LD_LIBRARY_PATH

> **Note:** Please refer the document oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

Compile Batch Libraries

To compile batch libraries, complete the following steps.

> **Note:** Verify that TNS is set up correctly by using the UP variable to successfully log in to the RMS 13 schema.
>
> **Example:** /u00/oracle> sqlplus $UP

1. Copy the files from BATCH_PATCH_DIR/batch-patch/13.2/oracle/lib/src to INSTALL_DIR/oracle/lib/src.
2. Change directories to INSTALL_DIR/oracle/lib/src.
3. To make library dependencies run this command.

   ```
   make –f retek.mk depend 2>&1 | tee libdpnd.log
   ```

   Check the libdpnd.log file for errors

4. To make batch libraries:
   ```
   make –f retek.mk retek rms resa 2>&1 | tee libretek.log
   ```
   Check the libretek.log file for errors
5. To install batch libraries:
   ```
   make –f retek.mk install
   ```
   The batch libraries should now be in INSTALL_DIR/oracle/lib/bin

Compile Batch Source Code

To compile batch source code, complete the following steps.

1. Copy the files from BATCH_PATCH_DIR/batch-patch/13.2/oracle/proc/src to INSTALL_DIR/oracle/proc/src.
2. Change directories to INSTALL_DIR/oracle/proc/src.
3. Create dependencies.
   a. Run the following command:
      ```
      make –f mts.mk depend 2>&1 | tee srcdpnd.log
      ```
   b. Check the srcdpnd.log file for errors.
4. Create batch programs.
   a. Run the following commands in the order stated.
      ```
      make –f rms.mk PRODUCT_PROCFLAGS=dynamic=ansi ditinsrt
          make –f mts.mk rms-ALL recs-ALL resa-ALL rtm-ALL fif-ALL 2>&1 | tee
      srcall.log
      ```
   b. Check the srcall.log file for errors.
5. Install the batch programs.
   ```
   make –f mts.mk install
   ```
   The batch programs should now be in INSTALL_DIR/oracle/proc/bin.
6. Copy RETL Code

   Copy the files from BATCH_PATCH_DIR/batch-patch/13.2/RETLfor<product> to INSTALL_DIR/RETLfor<product>.
7. Copy Data Conversion Scripts

   Copy the files from BATCH_PATCH_DIR/batch-patch/13.2/external to INSTALL_DIR/external.

3

Application Server Installation Tasks—Upgrade

There are two different methods to use for installing the RMS 13.2 Application. Option 1 uses the installer to apply upgrade. Option 2 compiles the RMS toolset and forms directly.

Option 1: Use Application Installer to Upgrade

The installer should only be used to apply the upgrade if the forms and libraries being upgraded do not contain customizations or hotfixes. If the upgrade is applied to customizations, they will be overwritten.

Before you apply the RMS 13.2 upgrade, do the following.

- Make a backup of all your forms and library files.
- Review the enclosed RMS 13.2 Upgrade Release Notes (rms-132-rn.pdf).
- Review each of the enclosed defect documents.

Before copying over any files, do the following.

- Note whether customizations have been made to the module. If so, then the customizations must be reapplied over the new version of the file (or the fix may need to be applied to the custom version of the code).
- Copy the original files to a different directory before installing in case they need to be referred to at a later date.

Run Application Installer

The steps to run the Application installer for an upgrade are the same as the steps for a full install. See the Application Server Installation Tasks—Full chapter for details.

In the installer screen titled WebLogic Configuration, the Configure WebLogic box should be unchecked. If you already have a working WebLogic forms environment, it does not need to be reconfigured. If you choose to check this option you may need to cleanup duplicate entries in the WebLogic formsweb.cfg file.

Test the RMS Application

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

Option 2: Compile RMS Toolset and Forms Directly

For Option 2, the RMS toolset and forms are compiled directly as follows.

Create Staging Directory for RMS Application Upgrade Files

To create a staging directory for RMS application upgrade files, complete the following steps.

1. Log into the application server as a user with read and write access to the Weblogic files.
2. Create a staging directory for the RMS application installation software. There should be a minimum of 350 MB disk space available in this location.
3. Copy the file rms1320apppatch.zip from the RMS 13.2 release to staging directory. This will be referred to as APP_PATCH_DIR when installing application software and reports.
4. Change directories to APP_PATCH_DIR and extract the file rms1320apppatch.zip.

Set Environment Variables

Notes:

INSTALL_DIR is the location where RMS 13 forms were installed.

ORACLE_HOME is the location where Oracle WebLogic (10.3.3) has been installed

ORACLE_INSTANCE is the location where WebLogic has been installed and contains the executables to compile forms

Make sure the following variables are set. The RMS 13.2 forms installer should have created a forms.profile file located at INSTALL_DIR/base/forms.profile. This profile script can be used to set all of the environment variables listed below.

Example: cd <INSTALL_DIR>/base

. ./forms.profile

Variables set by forms.profile:

- **All OS Platforms**
  - MMHOME=INSTALL_DIR/base
  - ORACLE_HOME=/path/to/WebLogic/as_1
  - ORACLE_INSTANCE=/path/to/WebLogic/asinst_1
  - ORACLE_SID= The Oracle Sid for the RMS database
  - UP=/@< Schema Owner Wallet Alias >
  - TNS_ADMIN=/path/to/wallet/files/dir/
  - NLS_LANG=AMERICAN_AMERICA.UTF8
  - DISPLAY=<IP address of X server>:0.0
  - 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/rms/forms/bin:$ORACLE_HOME/forms
  - TK_UNKNOWN=$ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
  - PATH=$ORACLE_INSTANCE/bin:$PATH

  Note: See the document oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

  Note: Verify that TNS is set up correctly by using the UP variable to successfully log in to the RMS 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 RMS Toolset

To install the RMS toolset, complete the following steps.

1. Make a backup copy of the existing INSTALL_DIR/base/toolset and INSTALL_DIR/base/forms directories.
2. Copy the contents of the RMS application patch into the above locations. APP_PATCH_DIR/app-patch/13.2/base/toolset into INSTALL _DIR/base/toolset, APP_PATCH_DIR/base/forms into INSTALL _DIR/base/forms, and APP_PATCH_DIR/app-patch/13.2/base/forms_scripts into INSTALL _DIR/base/forms_scripts
3. Copy all libraries (.pll files) in the INSTALL _DIR/base/toolset/src directory to the INSTALL _DIR/base/toolset/bin directory.
4. Change directories to INSTALL _DIR/base/toolset/bin.
5. Verify that the PATH variable contains the path INSTALL_DIR/base/forms_scripts. The forms.profile script should have set this up already.
6. Run toolset.pll.sh to compile all Toolset .pll's.

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

- messge45.pll
- ariiflib.pll
- stand45.pll
- calend45.pll
- find45.pll
- item45.pll
- tools45.pll
- mblock45.pll
- mview45.pll
- nav45.pll
- work45.pll
- itnumtype.pll
- hierfilter.pll
- rmslib.pll
- cflex.pll

7. Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully).
8. Remove all newly created .plx files.
9. Copy all forms (*.fmb files) in the INSTALL_DIR/base/toolset/src directory to the INSTALL_DIR/base/toolset/bin directory.
10. Run forms.fm_fmb.sh (in INSTALL_DIR/base/toolset/bin) to compile the Toolset reference forms.
11. Remove all newly created fm_*.fmx files (reference forms should not have executable files).
12. Run forms.fmb.sh (in INSTALL_DIR/base/toolset/bin) to generate Toolset runtime forms – .fmx's.
13. Check to make sure that each non-reference form (.fmb file) has a corresponding .fmx file.

    Note: Disregard fm_*.fmx files should they be created. These files should be removed. They should NOT exist in the INSTALL_DIR/base/toolset/bin directory.

14. Remove all non-reference form forms from INSTALL_DIR/base/toolset/bin; the following syntax leaves all reference forms (fm_*.fmb) in the bin directory, while removing all other forms:

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

15. Copy all menus (*.mmb files) in the INSTALL_DIR/base/toolset/src directory to the INSTALL_DIR/base/toolset/bin directory.
16. Run menus.mmb.sh (in INSTALL_DIR/base/toolset/bin) to generate Toolset runtime menus – .mmx's.

17. Check to make sure that each .mmb file has a corresponding .mmx file.

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

18. Remove all .mmb files from INSTALL_DIR/base/toolset/bin.

Install RMS Forms

To install RMS forms, complete the following steps.

1. Copy all the files from APP_PATCH_DIR/app-patch/13.2/base/forms/src to INSTALL_DIR/base/forms/src.
2. Copy all libraries (.pll files) in the INSTALL_DIR/base/forms/src directory to the directories to the INSTALL_DIR/base/forms/bin directory.
3. Change directories to INSTALL_DIR/base/forms/bin.
4. Run forms.pll.sh to compile all RMS .pll's.
5. Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully). Remove all newly created .plx files.
6. Copy all forms (*.fmb files) in the INSTALL_DIR/base/forms/src directory to the INSTALL_DIR/base/forms/bin directory.
7. Run forms.fm_fmb.sh (in INSTALL_DIR/base/rms/forms/bin) to compile the RMS reference forms.
8. Remove all newly created fm_*.fmx files (reference forms should not have executable files).
9. Run forms.fmb.sh (in INSTALL_DIR/base/rms/forms/bin) to generate RMS runtime forms – .fmx's.
10. Check to make sure that each non-reference form .fmb file has a corresponding .fmx file.

> **Note:** Disregard fm_*.fmx files should they be created. These files should be removed. They should NOT exist in the INSTALL_DIR/base/forms/bin directory.

11. Remove all non-reference form forms from INSTALL_DIR/base/forms/bin; the following syntax will leave all reference forms (fm_*.fmb) in the bin directory, while removing all other forms:

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

12. Copy all menus (*.mmb files) in the INSTALL_DIR/base/forms/src directory to the INSTALL_DIR/base/forms/bin directory.
13. Run menus.mmb.sh (in INSTALL_DIR/base/rms/forms/bin) to generate RMS runtime menus – .mmx's.
14. Check to make sure that each .mmb file has a corresponding .mmx file.
15. Remove all .mmb files from INSTALL_DIR/base/forms/bin.

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

Install Helpfile

To install the helpfile, complete the following steps.

1. Log into the WebLogic instance to which online help will be installed.
2. Create a server. In this example, rms-help-server is being used.
3. Select **Deployments**.
4. Click **Install**.
5. Click in the **Path:** box and enter APP_PATCH_DIR/app-patch/13.2/online-help/rms-help.ear, the ear file that will be deployed.
6. Leave **Install this deployment as an application** selected. Click **Next**.
7. Select the rms-help-server created in Step 2. Click **Next**.
8. Leave rms-help for the application name. Click **Next**.
9. Verify settings. Click **Finish**.

In the database, set the WEBHELP_SERVER column in the LANG table for the RMS schema owner to point to the RMS help server and port. For the above example, it is set to http://redevlv0065.us.oracle.com:17003.

4

RMS Reports Installation—Upgrade

RMS Reports are included in the RMS Application upgrade: rms132apppatch.zip in the reports directory. To install the reports files, copy them from the RMS application upgrade APP_PATCH_DIR/app-patch/13.2/reports to the reports directory created during RMS installation and repeat Step 4 in the RMS Reports—Full chapter to copy the reports files to Oracle BI Publisher.

5

Data Migration

Included in the 13.2 release is a tool responsible for upgrading preexisting data in the RMS schema to the 13.2 schema once 13.2 database upgrades are executed.

Before running the RMS 13.2 Data Migration Tool:

- Make a backup of all your objects and database schema.
- Check that RMS 13.2 is installed.
- Review the enclosed RMS 13.2 Upgrade Release Notes (rms-132-rn.pdf).
- Review each of the enclosed defect documents.

Create Staging Directory for RMS Data Migration Files

To create a staging directory for RMS data migration files, complete the following steps.

1. Log in to the database server as a user that can connect to the RMS database.
2. Create a staging directory for the RMS database schema installation software.
3. Copy the rms1320datamigration.zip file from the RMS 13.2 release to the staging directory. This is referred to as STAGING_DIR when running the data migration tool.
4. Change directories to STAGING_DIR and extract the rms1320datamigration.zip file. This creates a "master_controller" subdirectory under STAGING_DIR.

Configure RMS Data Migration Tool

To configure the RMS data migration tool, complete the following steps.

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

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

3. 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

4. Set and export the NLS_LANG environment variable.

   Example: NLS_LANG=AMERICAN_AMERICA.UTF8
   export NLS_LANG

5. Set and export the TNS_ADMIN environment variable.

   Example: TNS_ADMIN=/path/to/wallet/files/dir/

   export TNS_ADMIN

6. Open the controller.cfg file and replace the values for the following variables with the appropriate values.
   a. Export PATCH_DIR=STAGING_DIR/master_controller/rms
   b. export SCHEMA_OWNER=<The name of the RMS schema>
   c. export MMUSER=/@< Schema Owner Wallet Alias >

   Note: Please refer the document oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

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

   Example: /u00/oracle> sqlplus $UP

7. Configure the following files in the STAGING_DIR/master_controller/rms/files directory with data from your existing RMS schema for the migration. Use the existing files as templates for how this data should be formatted. For descriptions of this data, refer to the RMS 13.2 Data Model document (rms-132-dm.pdf).
   - state.dat

     state.dat is used to update the country for a state in the STATE table. This file is required if there are no stores/warehouses in the system or stores/warehouses are in more than one country. This check is done in the PREVALIDATION routine. Replace the default values in the template state.dat file with the correct values for your schema.
   - country_tax_jurisdiction.dat

     country_tax_jurisdiction.dat is used to populate the jurisdiction codes in the COUNTRY_TAX_JURISDICTION table. Replace the default values in the template country_tax_jurisdiction.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.
   - addr.dat

     addr.dat is used to update the jurisdiction code for a state/country in the ADDR table. Replace the default values in the template addr.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.
   - comp_store.dat

     comp_store.dat is used to update the jurisdiction code for a state/country in the COMP_STORE table. Replace the default values in the template comp_store.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

- competitor.dat

  competitor.dat is used to update the jurisdiction code for a state/country in the COMPETITOR table. Replace the default values in the template competitor.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

- comphead.dat

  comphead.dat is used to update the jurisdiction code for a state/country in the COMPHEAD table. Replace the default values in the template comphead.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

- customer.dat

  customer.dat is used to update the jurisdiction code for a state/country in the CUSTOMER table. Replace the default values in the template customer.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

- ordcust.dat

  ordcust.dat is used to update the jurisdiction code for a state/country in the ORDCUST table. Replace the default values in the template ordcust.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

- outloc.dat

  outloc.dat is used to update the jurisdiction code for a state/country in the OUTLOC table. Replace the default values in the template outloc.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

- rtv_head.dat

  rtv_head.dat is used to update the jurisdiction code for a state/country in the RTV_HEAD table. Replace the default values in the template rtv_head.dat file with the correct values for your schema. This is optional and required only if jurisdiction codes need to be loaded into the schema.

8. Run the following insert statement into your RMS schema manually. You can modify the default values if necessary:

```
insert into upg_item_supp_manu_country select item,supplier,origin_country_id,
'Y' from item_supp_country;
```

Run the RMS Data Migration Tool

Change directories to STAGING_DIR/master_controller/rms.

1. If rerunning the data migration process, clear the contents of the "processed" directory.
2. Run prevalidation tool. This ensures that the input files for the data migration tool is up to date:

```
$ ./rms132_upgrade.ksh PREVALIDATION
```

3. Run migration tool.

```
$ ./rms132_upgrade.ksh UPGRADE
```

4. Run migration cleanup tool. This removes temporary data migration objects from the database.

   ```
   $ ./rms132_upgrade.ksh CLEANUP
   ```

5. Refer to the files in the log and error directory for details if there are problems during migration.
6. You will need to rebuild synonyms for any additional RMS users. See the Set Up Additional RMS Users section how to create synonyms.

Configure ReIM Data Migration Tool

If you choose to migrate ReIM data, follow these steps.

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

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

3. 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

4. Set and export the NLS_LANG environment variable.

   Example: NLS_LANG=AMERICAN_AMERICA.UTF8
   export NLS_LANG

5. Set and export the TNS_ADMIN environment variable.

   Example: TNS_ADMIN=/path/to/wallet/files/dir/
   export TNS_ADMIN

6. Open the controller.cfg file and replace the values for the following variables with the appropriate values:
   a. Export PATCH_DIR=STAGING_DIR/master_controller/reim
   b. export SCHEMA_OWNER=<The name of the RMS schema>
   c. export MMUSER=/@< Schema Owner Wallet Alias >

   Notes:

   See the document "oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

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

   Example: /u00/oracle> sqlplus $UP

Run the ReIM Data Migration Tool

To run the ReIM data migration tool, complete the following steps.

1. Change directories to STAGING_DIR/master_controller/reim.
2. If rerunning the data migration process, clear the contents of the processed directory.
3. Run migration tool.
   ```
   $ ./reim132_upgrade.ksh UPGRADE
   ```
4. Run migration cleanup tool. This removes temporary data migration objects from the database.
   ```
   $ ./reim132_upgrade.ksh CLEANUP
   ```
5. Refer to the files in the log and error directory for details if there are problems during migration.
6. Rebuild synonyms for any additional RMS users. See the section, Set Up Additional RMS Users, for how to create synonyms.

Configure Allocation Data Migration Tool

If you choose to migrate Allocation data, follow these steps:

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

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

3. 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

4. Set and export the NLS_LANG environment variable.

   Example: NLS_LANG=AMERICAN_AMERICA.UTF8
   export NLS_LANG

5. Set and export the TNS_ADMIN environment variable.

   Example: TNS_ADMIN=/path/to/wallet/files/dir/
   export TNS_ADMIN

6. Open the controller.cfg file and replace the values for the following variables with the appropriate values :
   a. Export PATCH_DIR=STAGING_DIR/master_controller/alloc-rms
   b. export SCHEMA_OWNER=<The name of the RMS schema>
   c. export MMUSER=/@< Schema Owner Wallet Alias >

   Note: Please refer the document oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

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

   Example: /u00/oracle> sqlplus $UP

Run the Allocation Data Migration Tool

To run the allocation data migration tool, complete the following steps.

1. Change directories to STAGING_DIR/master_controller/alloc-rms.
2. If rerunning the data migration process, clear the contents of the processed directory.
3. Run prevalidation tool. This ensures that the input files for the data migration tool is up to date:
   ```
   $ ./allocation132_upgrade.ksh PREVALIDATION
   ```
4. Run migration tool.
   ```
   $ ./allocation132_upgrade.ksh UPGRADE
   ```
5. Run migration cleanup tool. This removes temporary data migration objects from the database.
   ```
   $ ./allocation132_upgrade.ksh CLEANUP
   ```
6. Refer to the files in the log and error directory for details if there are problems during migration.
7. Rebuild synonyms for any additional RMS users. See the section, Set Up Additional RMS Users, for how to create synonyms.

6

Web Services Installation

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

Extract Web Services Files

To extract Web Services files, do the following:

1. Create a directory under the Batch INSTALL_DIR to hold the web services files. This will be referred to as Web Service Objects.
2. Copy the rms1320webservices.zip file from the RMS 13.2 release to INSTALL_DIR/'Web Service Objects'.
3. Unzip rms1320webservices.zip from INSTALL_DIR/'Web Service Objects'.

Create a Managed Server

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

Create a Datasource

Create a datasource for RMS Webservices which points to the RMS schema as follows.

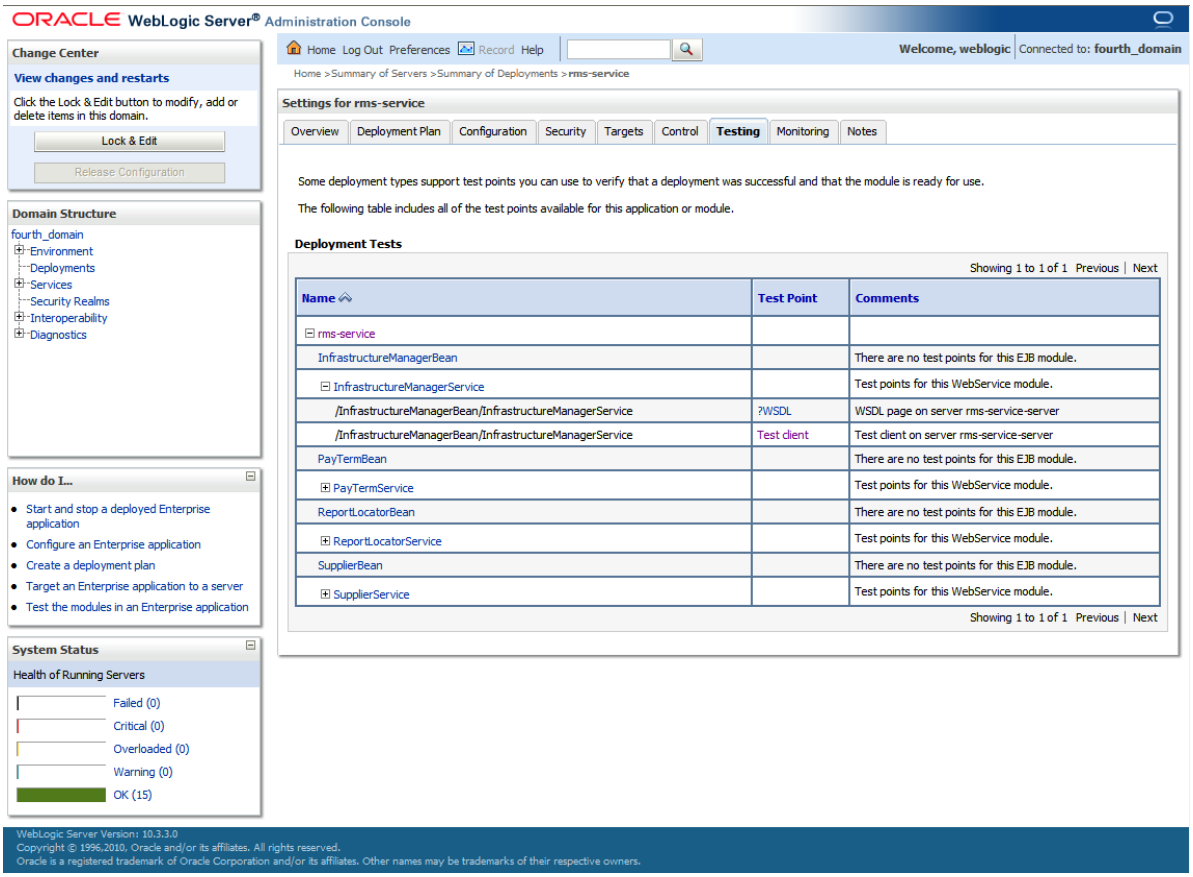
- Name can be anything you want.
- JNDI Name must be jdbc/RetailWebServiceDs.
- Set database type and driver for your environment.
- Set connection properties for the database using the rms user (rms01user). Be sure to test the configuration before moving on.
- Point the data source to the server created in the Create a Managed Server section above.

Deploy RMS Service EAR File

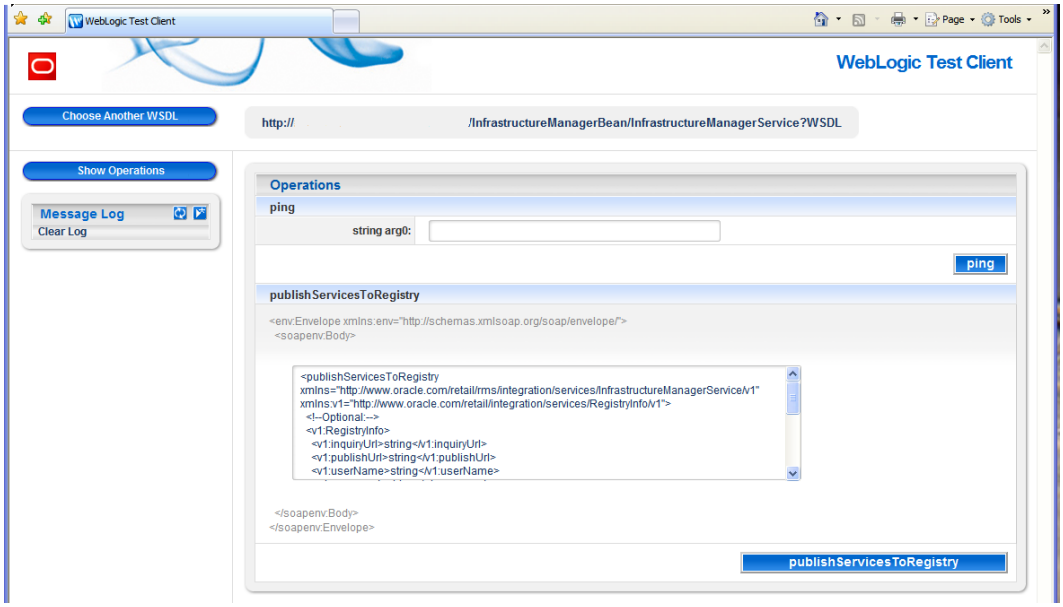
To deploy the RMS Service .ear file, do the following.

1. Make sure that the managed server created in Step 2, where this application will be deployed, is up and running.
2. In the left Domain Structure window, click Environment > Deployments.
3. Click **Lock and Edit** in the change center to install the ear file. It will enable the install button on the deployments screen.
4. Click Install.
5. Click the **upload your file(s)** link.
6. Click the **Deployment Archive** browse button.
7. Select the rms-service.ear file from local machine.
8. Click **Next**. Make sure that the radio button for rms-services.ear is selected.

9. Click **Next** again. Make sure that **Install this deployment as an application** is selected.
10. Click **Next** again and select the server created in Step 2.
11. Click **Next**. Click **Finish** to return to the deployments page. You should see rms-service in the list of deployments.
12. Click **Activate Changes** in the change center. The state of the application may be shown as prepared. If so, select the check box next to rms-service to will enable the **Start** button. Click **Start**. Select **servicing all requests**.
13. To test the deployment, click on the application. Click the testing tab.



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



Configure Web Service Security

> **Note:** If you are utilizing the AIA 2.5 solution for PeopleSoft or EBS, you must skip this section.

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

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

> **Note**: the AIA 2.5 solution for PeopleSoft and EBS does not support the WS-Security Username profile.

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

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

These steps are explained below.

Attach Policy File to the Web Service

The ear file contains **usernametoken.xml** in the **META-INF/policies** folder. This file contains the policy which is used by the web service. Follow the below steps to attach it to a web service:

1. Click on the application in the deployments screen.
2. The overview page of the application shows all the modules and components which are installed as part of the application:
3. Click on the web service for which you want to enable security.
4. Click the Configuration > WS-Policy tab of the web service. You should see the Web Wervice port (for example, PayTermPort) in the list of service endpoints.
5. Click the plus sign next to the Web Service to show all the operations of the Web Service.
6. Secure all the operations of the Web Service or selected operations of the Web Service. If you click on the Web Service port, it will show the next page, where you can attach policy file to the web service.
7. In the Available Endpoint Policies list, select the policy:usernametoken.xml option and move it to the list of Chosen Endpoint Policies.
8. Click **OK**. Leave all default values as they are.
9. Click **OK** again.
10. Verify the policy details have been added by clicking on the Testing tab of the web service page, and then selecting the WSDL. The WSDL should contain content similar to that shown below, along with additional service specific details:

```
<?xml version='1.0' encoding='UTF-8'?>
 <definitions xmlns:wssutil="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.oracle.com/retail/rms/integration/services/PayTermServic
e/v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://www.oracle.com/retail/rms/integration/services/PayTerm
Service/v1" name="PayTermService">
<wsp:UsingPolicy wssutil:Required="true" />
    <wsp:Policy wssutil:Id="usernametoken">
    <ns1:SupportingTokens xmlns:ns1="http://docs.oasis-open.org/ws-sx/ws-
    securitypolicy/200512">
    <wsp:Policy>
<ns1:UsernameToken ns1:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200512/IncludeToken/AlwaysToRecipient">
<wsp:Policy>
<ns1:WssUsernameToken10 />
</wsp:Policy>
</ns1:UsernameToken>
</wsp:Policy>
</ns1:SupportingTokens>
</wsp:Policy>
```

Create Roles and Users who can Access the Web services

The second step is to create roles and users who can access the Web services, as follows.

1. Add users to the security realm. In the Domain Structure window, click the Security Realms link. The default realm is shown.
2. Click the link on the realm.
3. Click the Users and Groups tab.
4. Click **New**. And enter user name and password details on the next screen. Leave the default value for Provider.
5. Click **OK** to save the changes. The new user is shown in the list of users
6. Add roles either from the Roles and Policies tab of the security realm, or through the Security tab of the Web Wervice. This example shows how to create a role from the security tab of the Web Wervice.
7. Navigate to the Security tab of the Web Service to which you are adding security.
8. In the Roles tab, click **New**.
9. In the Name field, enter the role name. For example, rmsrole. Leave the Provider Name to default value. Click **OK**. The newly created role is shown in the role tab.
10. Add the user to the role. Click on the newly created role.
11. Click **Add Conditions**.
12. Select User in the Predicate List drop down. Click **Next**.
13. Enter the user name that was created in the security realm. Click **Add**. It is added to the list below the text box.
14. Click **Finish**.
15. Click **Save**.
16. Navigate back to the Security > Policies tab of the Web Service.
17. Click **Add Conditions**.
18. Select Role in the Predicate List drop down. Click **Next**.
19. Enter the role name that was created earlier and click **Add**. The role is added in the list below the text box.
20. Click **Finish**.
21. Click **Save**.
22. The process of adding security to Web Services is complete. Open the test page of the Web Service and verify that the Web Service is secured.

A

Appendix: Oracle 11g Database Parameter File

```
###########################################################################
# Oracle 11.2.0.x Parameter file
#
# NOTES: Before using this script:
#        1.  Change <datafile_path>, <admin_path>, <utl_file_path>, <diag_path>
and <hostname>
#            values as appropriate.
#        2.  Replace the word SID with the database name.
#        3.  Size parameters as necessary for development, test, and production
environments.
# ----------------------------------------------------------------------
# MAINTENANCE LOG
#
# Date     By          Parameter           Old/New         Notes
# +------+ +---------+ +-----------------+ +-------------+ +-------------+
#
#
###########################################################################
# ----------------------------------------------------------------------------
# The policy is to give 60% for sga and 40% for PGA out of Memory Target at
startup
# ----------------------------------------------------------------------------
memory_target                          = 2000M
# ----------------------------------------------------------------------------
audit_file_dest              = <admin_path>/adump
compatible            = 11.2.0
control_files                            = (<datafile_path>/control01.ctl
                                           ,<datafile_path>/control02.ctl)
db_block_size             = 8192      # Default is 2k; adjust before db creation,
cannot change after db is created
db_file_multiblock_read_count        = 16        # Platform specific (max io
size)/(block size)
db_name               = SID
diagnostic_dest                          = '<diag_path>'
java_pool_size            = 100M
job_queue_processes           = 5         # Oracle Retail required; number of
cpu's + 1
local_listener                =
"(ADDRESS=(PROTOCOL=TCP)(HOST=<hostname>)(PORT=1521))"
nls_calendar                             = GREGORIAN
nls_date_format               = DD-MON-RR # Oracle Retail required; if RDW
database see later entry for proper format
nls_language              = AMERICAN  # Default
nls_numeric_characters        = ".,"      # Should be explicitly set to ensure all
users/batch get the same results
nls_sort              = BINARY    # Should be explicitly set to ensure all
sessions get the same order
nls_territory             = AMERICA   # Default
open_cursors              = 900       # Oracle Retail required (minimum=900);
default is 50
plsql_optimize_level                     = 2         # 10g change; use this setting
to optimize plsql performance
```

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

B

Appendix: Configure Listener for External Procedures

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

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

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

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (PROGRAM = extproc)
      (SID_NAME = extproc_agent)
      (ENVS='EXTPROC_DLLS=ANY')
    )
)
```

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

```
########################################################################
# File: tnsnames.ora
# Desc: Net Services configuration file.
# Note: Change these values: <service_name>, <oracle_sid>, <hostname>,
#       <global_name>
########################################################################

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

EXTPROC_CONNECTION_DATA.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

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

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

Example:
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

EXTPROC_CONNECTION_DATA.world =
  (DESCRIPTION =
    (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(Key = extproc_key)))
    (CONNECT_DATA = (SID = extproc_agent)))

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

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

C

Appendix: Tablespace Creation Scripts

```
------------------------------------------------------------------------------
---
--- Script:        create_rms_tablespaces.sql
--- Execute as:    sysdba
--- Note:          Before running this script:
---                  Modify <datafile_path> values.
---                  Modify datafile storage parameters and sizes based on
partitioning strategy.
------------------------------------------------------------------------------
spool create_rms_tablespaces.log

CREATE TABLESPACE RETEK_INDEX
DATAFILE '<datafile_path>/retek_index01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE RETEK_DATA
DATAFILE '<datafile_path>/retek_data01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE LOB_DATA
DATAFILE '<datafile_path>/lob_data01.dbf'  SIZE 50M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE USERS DATAFILE
    '<datafile_path>/users01.dbf'  SIZE 100M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;

ALTER TABLESPACE RETEK_INDEX
ADD DATAFILE '<datafile_path>/retek_index02.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;
ALTER TABLESPACE RETEK_INDEX
ADD DATAFILE '<datafile_path>/retek_index03.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;
ALTER TABLESPACE RETEK_INDEX
ADD DATAFILE '<datafile_path>/retek_index04.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;
ALTER TABLESPACE RETEK_INDEX
ADD DATAFILE '<datafile_path>/retek_index05.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;
ALTER TABLESPACE RETEK_INDEX
ADD DATAFILE '<datafile_path>/retek_index06.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;
```

```
ALTER TABLESPACE RETEK_DATA
ADD DATAFILE '<datafile_path>/retek_data02.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;
ALTER TABLESPACE RETEK_DATA
ADD DATAFILE '<datafile_path>/retek_data03.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
;


spool off

exit
```

D

Appendix: RMS User Creation Script

Run the following commands as the sysdba user. Replace **schema_owner** with an appropriate account name. The empty role developer must be created before running the following commands.

```
accept schema_owner prompt 'Please enter the schema owner name: '
accept password prompt 'Please enter the password for the schema owner: '
accept temptblsp prompt 'Please enter the temporary tablespace for the schema
owner: '


create user &schema_owner
identified by &password
default tablespace RETEK_DATA
temporary tablespace &temptblsp
quota unlimited on RETEK_DATA
quota unlimited on RETEK_INDEX
quota unlimited on LOB_DATA
/

spool create_&schema_owner

grant developer,
      select_catalog_role,
      alter session,
      analyze any,
      create any synonym,
      create any type,
      create database link,
      create library,
      create procedure,
      create sequence,
      create session,
      create synonym,
      create table,
      create trigger,
      create view,
      create job,
      drop any synonym,
      execute any procedure,
      execute any type,
      select any sequence,
      select any table,
      query rewrite,
      create materialized view,
      create any context to &schema_owner
/

grant select on sys.dba_role_privs to &schema_owner
/

grant select on sys.dba_jobs to &schema_owner
/

grant select on sys.dba_roles to &schema_owner
/
```

```
grant select any dictionary to &schema_owner
/

grant select on dba_jobs to &schema_owner with grant option;
grant select on dba_roles to &schema_owner with grant option;
grant select on dba_role_privs to &schema_owner with grant option;
grant execute on dbms_rls to &schema_owner with grant option;
grant execute on dbms_alert to &schema_owner;
grant execute on dbms_pipe to &schema_owner;
grant select_catalog_role to &schema_owner;
grant execute_catalog_role to &schema_owner;
grant execute on dbms_lock to &schema_owner;
grant execute on dbms_rls to &schema_owner;
grant execute on dbms_crypto to &schema_owner;
grant query rewrite to &schema_owner;
grant select on dba_sys_privs to &schema_owner with grant option;
grant execute on sys.dbms_system to &schema_owner;
--
-- aq related privileges
--
grant execute on dbms_aq to &schema_owner;
grant execute on dbms_aqadm  to &schema_owner;
grant aq_administrator_role  to &schema_owner;
exec dbms_aqadm.grant_system_privilege(privilege => 'DEQUEUE_ANY', GRANTEE =>
'&schema_owner', ADMIN_OPTION => FALSE);
exec dbms_aqadm.grant_system_privilege(privilege => 'ENQUEUE_ANY', GRANTEE =>
'&schema_owner', ADMIN_OPTION => FALSE);
exec dbms_aqadm.grant_system_privilege(privilege => 'MANAGE_ANY', GRANTEE =>
'&schema_owner', ADMIN_OPTION => FALSE);


spool off
```

E

Appendix: RMS RETL Instructions

This Appendix summarizes the RETL program features utilized in the RMS Extractions (RMS ETL). More information about the RETL tool is available in the latest RETL Programmer's Guide. More information about RMS ETL is available in the RMS ETL operations guide.

Configuration

RETL

Before trying to configure and run RMS ETL, install RETL version 10.3 or later which is required to run RMS ETL. Run the "verify_retl" script (included as part of the RETL installation) to ensure that RETL is working properly before proceeding.

RETL User and Permissions

RMS ETL should be installed and run as the RETL user. Additionally, the permissions should be set up as per the RETL Programmer's Guide. RMS ETL reads data, creates, deletes and updates tables. (This is to ensure that weekly sales data is not pulled multiple times on subsequent extractions.) If these permissions are not set up properly, extractions will fail.

Environment Variables

In addition to the RETL environment variables (please see the Programmer's Guide for version of RETL), you need to set MMHOME to the base directory for RMS ETL. This is the top level directory that selected during the RMS Batch installation process. So in .kshrc you should add a line like the following:

```
export MMHOME=<base directory for RMS ETL>
```

rmse_config.env

There are a couple variables that will need to change depending upon local settings:

```
export DBNAME=int9i
export RMS_OWNER=RMS13DEV
export BA_OWNER=rmsint1012
```

Also, you will need to set the environment variable PASSWORD in either the rmse_config.env, .kshrc or some other location that can be included via one of those two means. For example, adding this line to the rmse_config.env will cause the password, bogus, to be used to log into the database: `export PASSWORD=pass1`

F

Appendix: Oracle Trade Management System Expectations

This appendix describes the items expected by the Oracle Trade Management System.

Installation Scripts (elc_comp_post_htsupld.sql)

This script is for the RTM product only. This needs to be applied only after all static install scripts have been run, oga, tariff_treatment, quota_category, country_tariff_treatment and hts_headings scripts have all been run followed by running the htsupld.pc program. The last step is running this script. This script inserts the Expense and Assessment Cost Components. This script needs to be run once for each country of import that the client is using.

> **Note:** This script is expecting two parameters to be passed in (the user will be prompted for the parameters). The first parameter is country ID, this is the Import Country. The second parameter is Currency Code, this is the code of the currency that corresponds to the entered Import Country. Most likely this script will be run using the Base Country and the Primary Currency as defined in the System Variables form.

The inserted components include:

- MPFXX (Merchandise Processing Fee XX) – This component is used to store Merchandise Processing Fee. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one component created, MPFUS, with a description of Merchandise Processing Fee US. This leaves the client with the ability to create additional MPF components for each of the countries that they intend to import into. This component is inserted with a Component Rate of 100 percent. This rate should be modified to be the appropriate rate for the Import Country. This component is also set up as an Always Default which means that it is defaulted to every Item/HTS combination.
- HMFXX (Harbor Maintenance Fee XX) – This component is used to store Harbor Maintenance Fee. In place of the XX will be the country code that is passed into the script. So if the Country is US, then there is one component created, HMFUS, with a description of Harbor Maintenance Fee US. This leaves the client with the ability to create additional HMF components for each of the countries that they intend to import into. This component is inserted with a Component Rate of 100 percent. This rate should be modified to be the appropriate rate for the Import Country.
- TDTYXX (Total Duty XX) – This component is used to store the total of the duty for each Item/HTS or Order/Item/HTS combination. It totals all duties, taxes, and fees within the Ordering dialog. This total is added together with the Total Expense and the Item's Cost to come up with the Total Estimated Landed Cost of the Item or Order/Item combination. This component should not be modified.

- VFDXX (Value For Duty XX) – This Computation Value Base (CVB) is used to store the value that duty should be calculated from. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one CVB created, VFDUS, with a description of Value for Duty US. This leaves the client with the ability to create additional VFD CVBs for each of the countries that they intend to import into. Upon insert here, this CVB will only have one detail, which is ORDCST (Order Cost). If the client needs additional expenses (we are making the assumption that only Expense components will make up Value for Duty) to be used in the Value For Duty, they need to be added to VFDXX through SQL Plus. All automatically inserted Assessment components with a Calculation Basis of Value will have VFDXX as the CVB.
- VFDXXXX (XX% of Value For Duty XX) – This component is used to store a percent of the CVB, Value For Duty. This is used in the case when you have an Item that is classified with multiple HTS codes. For example, a button-down shirt may have one HTS code for the cotton material that is 75 percent of the cost, and a second HTS code for the buttons that make up the other 25 percent. The duty components associated with the first HTS code would be need to be calculated from 75 percent of the entire Value for Duty. To accomplish this, the associated components would use VFD75XX as their CVB instead of VFDXX. The detail component would be 'VFD75XX' and would have a Component Rate of 75 and a CVB of VFDXX, therefore, the component VFD75XX would be 75% of the Value for Duty. More generically speaking, VFDXXXX will be the only detail in an inserted CVB called VFDXXXX, where the first XX is replaced with the percentage. In place of the second XX will be the country code that is passed into the script. So if the Country is US, then there will be one component created, VFD25US, with a description of 25% of Value for Duty US. This leaves the client with the ability to create additional VFD components for each of the countries that they intend to import into. The script will insert VFD25XX, VFD50XX, and VFD75XX, these are meant to be used as a guide if the client needs additional components with different percentages. These components should not be modified.
- DTYXXXX (DTYXXXX) – These components are used to calculate duty for each HTS code. In place of the first XX is the HTS code's Duty Component Code concatenated with an A, B, or C as needed for duty calculation. In place of the second XX is the country code that is passed into the script. So if the Country is US, then there is one component created, DTYXXUS, with a description of DTYXXUS. This leaves the client with the ability to create additional components for each of the countries that they intend to import into. The Import Country for these components is the country code of the Base Country that is defined on the System Options table. This component is inserted with a Component Rate of 100 percent. This rate is overwritten with the appropriate Tariff Treatment rate upon calculation within the Item and Ordering dialogs. These components should not be modified.
- DUTYXX(DUTYXX) – This component is used as a sub-total. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one component created, DUTYUS, with a description of DUTYUS. This leaves the client with the ability to create additional components for each of the countries that they intend to import into. It contains the sum of all DTYXXXX components each HTS code. This component has a CVB called DUTYXX that contains every 'DTYXXXX' component as its details. This component should not be modified.

- XXXXXXX (XXXXXXX) – Fees and Taxes are created using a concatenation of information. The Component ID consists of the Fee or Tax Class Code concatenated with the Fee or Tax Component Code, and an A or B as needed for calculation, and then the import country. For example, there is an existing Fee Class Code (also referred to as Fee Type) which is 053, its Fee Component Code is 1, and importing into the US, so there is a component created that has an ID of 0531AUS. The descriptions are the same as the Component ID and can/should be modified to be clearer. Other than the description, these components should not be modified.
- ADXX (Anti-Dumping XX) – This component contains the Anti-Dumping charge for each Item/HTS code. In place of the XX is the country code that is passed into the script. So if the Country is US, then there is one component created, ADUS, with a description of Anti-Dumping US. This leaves the client with the ability to create additional components for each of the countries that they intend to import into. This component should not be modified.
- CVDXX (Countervailing Duty XX) – This component contains the Countervailing Duty charge for each Item/HTS code. In place of the XX will be the country code that is passed into the script. So if the Country is US, then there is one component created, CVDUS, with a description of Countervailing Duty US. This component should not be modified.

HTS Upload / Mass Update

There are several installation scripts that must be run prior to HTS Upload to populate the following tables. These are one-time installations upon implementation of the product and must be maintained by the client.

- ELC_COMP
- QUOTA_CATEGORY (through the quota_category.sql script)
- OGA (through the oga.sql script)
- COUNTRY_TARIFF_TREATMENT (via the country_tariff_treatment.sql script)
- HTS_CHAPTER (via the hts_headings.sql script)
- TARIFF_TREATMENT (through the tariff_treatment.sql script)

After the initial load of the HTS data from executing the HTS Upload program. One additional install script must be run to populate the following tables with additional information:

- ELC_COMP, CVB_HEAD, CVB_DETAIL (through the elc_comp_post_htsupld.sql script)

The initial load of HTS information using a Customs provided tape and subsequent execution of the HTS Upload program will populate and update the following tables:

- HTS
- HTS_TARIFF_TREATMENT
- HTS_OGA
- HTS_FEE
- HTS_TAX
- HTS_TT_EXCLUSIONS

The following tables need to be populated by the client, but will be updated through the HTS Upload program.

- HTS_AD
- HTS_CVD
- HTS_REFERENCE

The following tables need to be populated and maintained by the client:

- HTS_CHAPTER_RESTRAINTS

Calculation of Merchandise Processing Fee

This particular cost component is the only Cost Component that is calculated with a Min/Max Range for each Customs Entry. This range is defined on the MPF_MIN_MAX table (note: this table does not have a corresponding form and needs to be populated by the client via SQL Plus. In order to process MPF the MPF_MIN_MAX table must be populated for the import country or else the calculation function errors out during processing.). If a client does not use Merchandise Processing Fee, but has a similar component, they can use the MPF_MIN_MAX table and the MPFXX component to accomplish the same result. They simply need to change the Component Description and Rate. Within the Customs Entry dialog, MPFXX is defaulted in along with all other assessments that are associated with each Order/Item combination. Once associated with the Entry, MPF is recalculated and checked to see if the value falls within the Min/Max Range. If not, the value is modified to be within the range and then allocated across all of the items on the Entry. Because this value is being calculated by the system, the user is not allowed to modify the rate or value of any MPF components within the Customs Entry dialog.

Unit of Measure Conversions

The internal process that calculates and distributes MPF charges on-line requires Unit of Measure (UOM) conversions in multiple instances. If a particular UOM conversion is missing the processing stops and a message will be displayed indicating that there is insufficient UOM information to continue. If this should occur, you must exit the dialog that generated the error add the missing conversion information and re-enter the dialog for the MPF charges to be processed.

Customs Entry Ref. Status

There are 4 possible CE Ref. Statuses for each Customs Entry. They are Worksheet, Send, Downloaded, and Confirmed. In general when an Entry is created it is in Worksheet status. Once all of the necessary information has been added, the user is set the Status to Send, indicating that the Entry is ready to be sent to the Broker. That night in the nightly batch run, the Entry is downloaded to the Broker (cednld.pc). Once the download process is complete, the Status is automatically set to Downloade; a user can never set the Status to this value manually. At that point once the user receives confirmation from the Broker, makes any necessary changes, and is sure that the information is correct, they can set the CE Ref. Status to 'Confirmed'. From that point on the Status cannot be changed, however most of the fields on the CE Header form remain editable. All information on the CE Shipment form is view only. Also, all information on the CE Order/Item form is view only except for the Cleared Quantity, Cleared Quantity UOM, Apply button, and Comments fields. And finally all information in the CE Charges form will be view only as well.

Since some clients may prefer not to download their Entries to a Broker, the user will have the ability to set the CE Ref. Status from Worksheet directly to Confirmed.

Customs Entry Totals

The following describes customs entry totals.

- Total Duty contains the sum of the duty charges (any component beginning with DTY) for each item times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Taxes contains the sum of the tax charges (any component beginning with a tax type (see attached document for a description of taxes)) for each item times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Other contains the sum of all other charges (including fees) for each item times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total VFD contains the Value for Duty (which can be made up of order cost plus other dutiable expenses such as selling commission, royalties, etc.) times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Est. Assessments contains the sum of the estimated duty/fees/taxes for each item, calculated from the Purchase Order/Item HTS Assessments, times the associated item's Manifest Item quantity, summed together for all items on the entry.
- Total Act. Assessments contain the sum of the Total Duty, Total Taxes, and Total Other values.

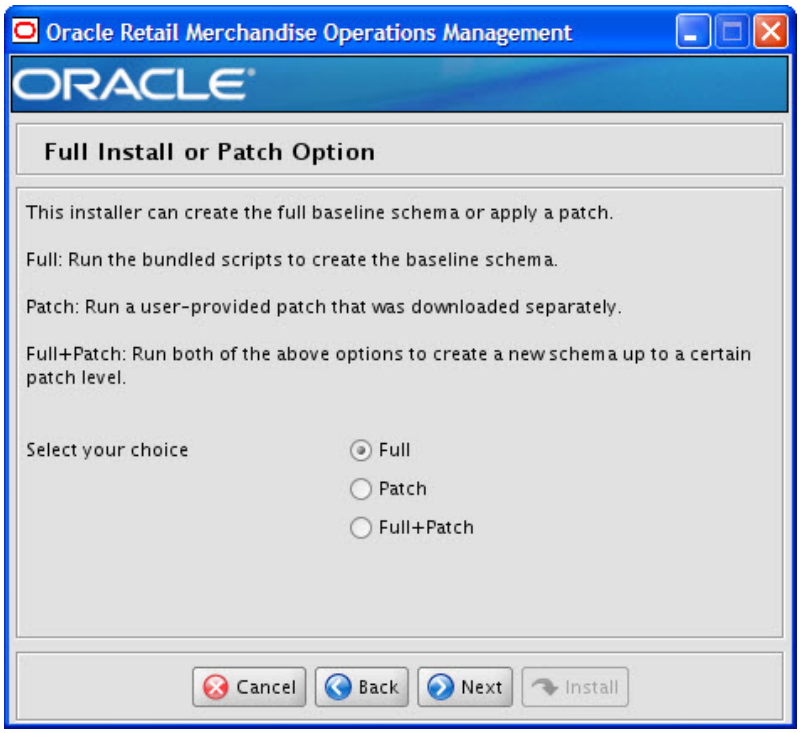
G

Appendix: RMS Database Schema Installer Screens

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

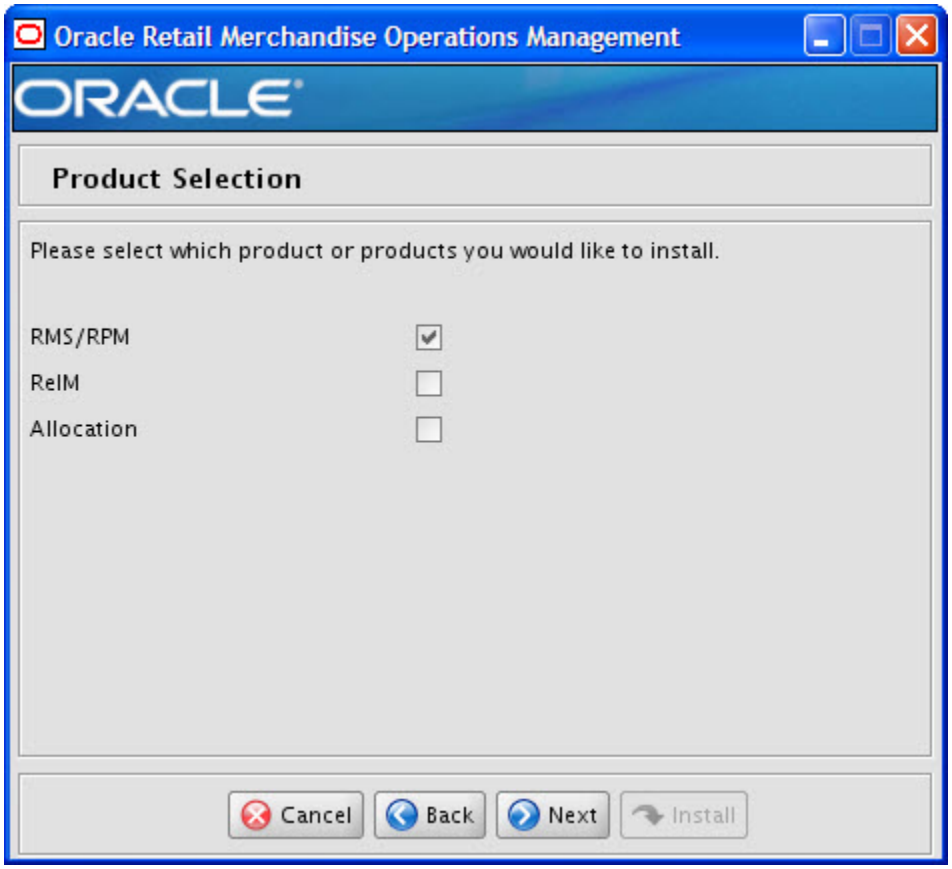
The RMS database schema installer also includes the option to install the database schema objects for the ReIM and Allocation products.

Screen: Full Install or Patch Option



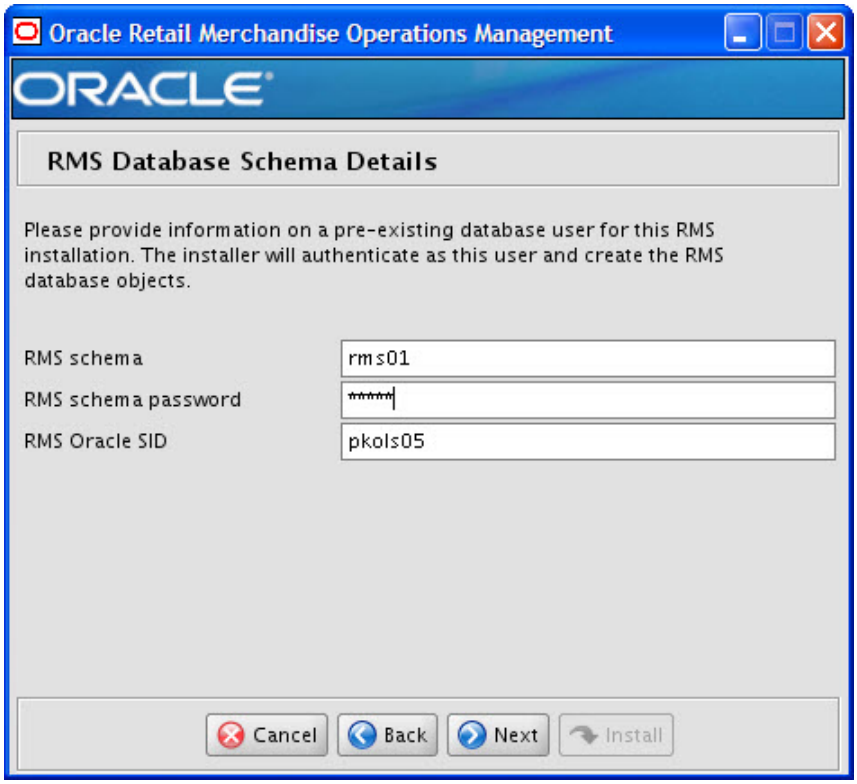
Field Title	Full or Patch
Field Description	The installer can create the full baseline schema, apply a patch, or do both. To install a new instance of the RMS 13.2 release, select Full. If installing 13.2 upgrade or later, select Patch and the installer prompts for the location of the patch files on the next screen.
Example	Full

Screen: Product Selection



Field Title	Product Selection
Field Description	By default the RMS database schema installer creates the database objects for RMS/ReSA/RTM and RPM. Optionally, the database objects for ReIM and/or Allocation may be installed at the same time or later.
Example	RMS/RPM, ReIM, Allocation

Screen: RMS Database Schema Details



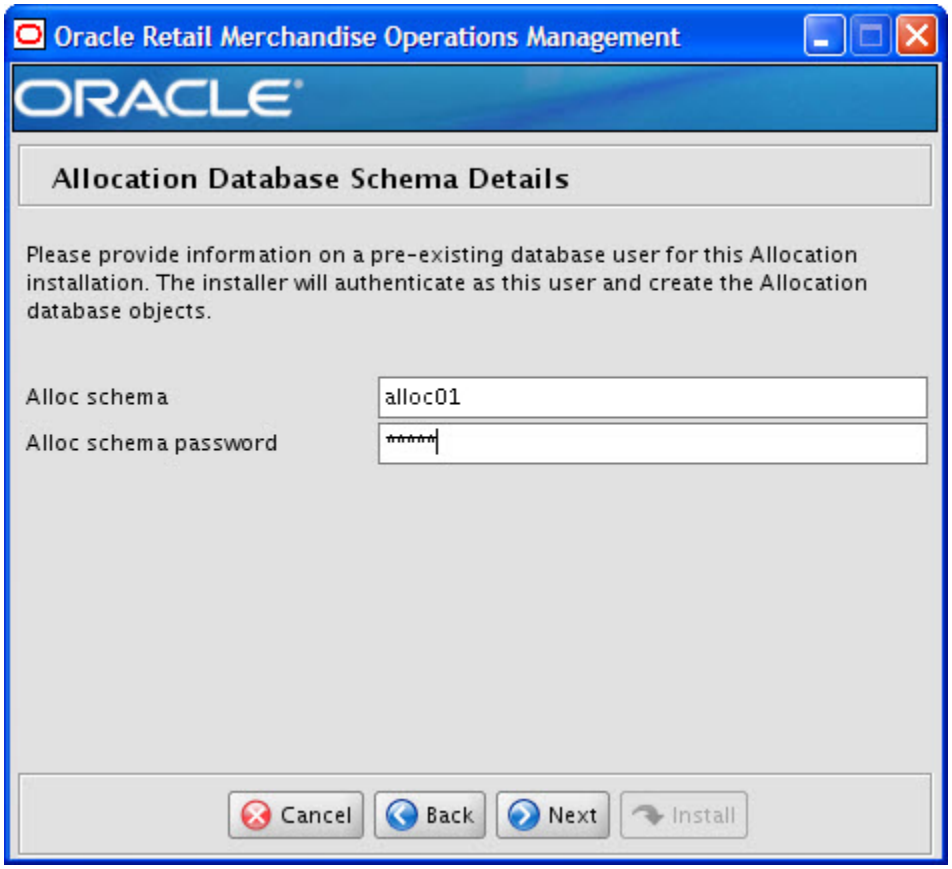
Field Title	RMS schema
Field Description	Provide the RMS database user here. The installer logs into the database as this user to create the RMS schema. This user must already exist in the database when the RMS database schema installer is run.
Example	rms01

Field Title	RMS schema password
Field Description	Database password for the RMS Schema Owner.

Field Title	RMS Oracle SID
Field Description	Oracle system identifier for the database where RMS will be installed
Example	pkols05

The database settings provided are validated by the installer when you advance to the next screen.

Screen: Allocation Database Schema Details

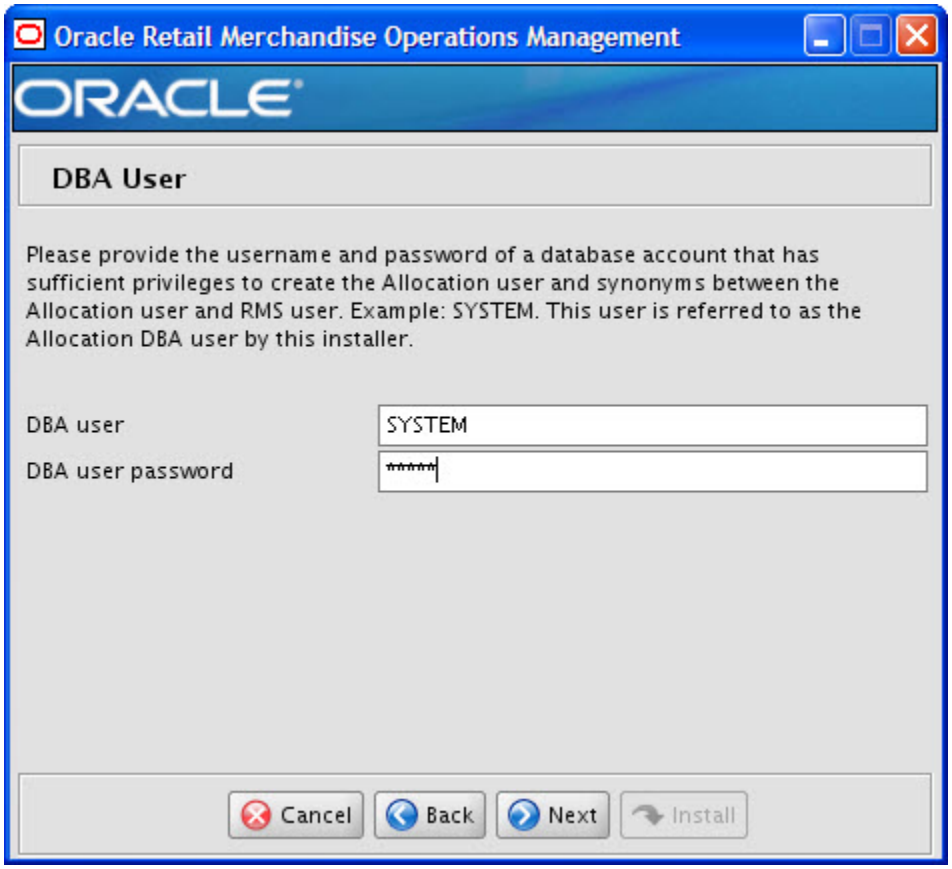


Field Title	Alloc schema
Field Description	Provide the Allocation database user here. The installer logs into the database as this user to create the Allocation schema objects. This user must already exist in the database when the database schema installer is run.
Example	alloc01

Field Title	Alloc schema password
Field Description	Database password for the Allocation database user.

The database settings provided are validated by the installer when you advance to the next screen.

Screen: DBA User

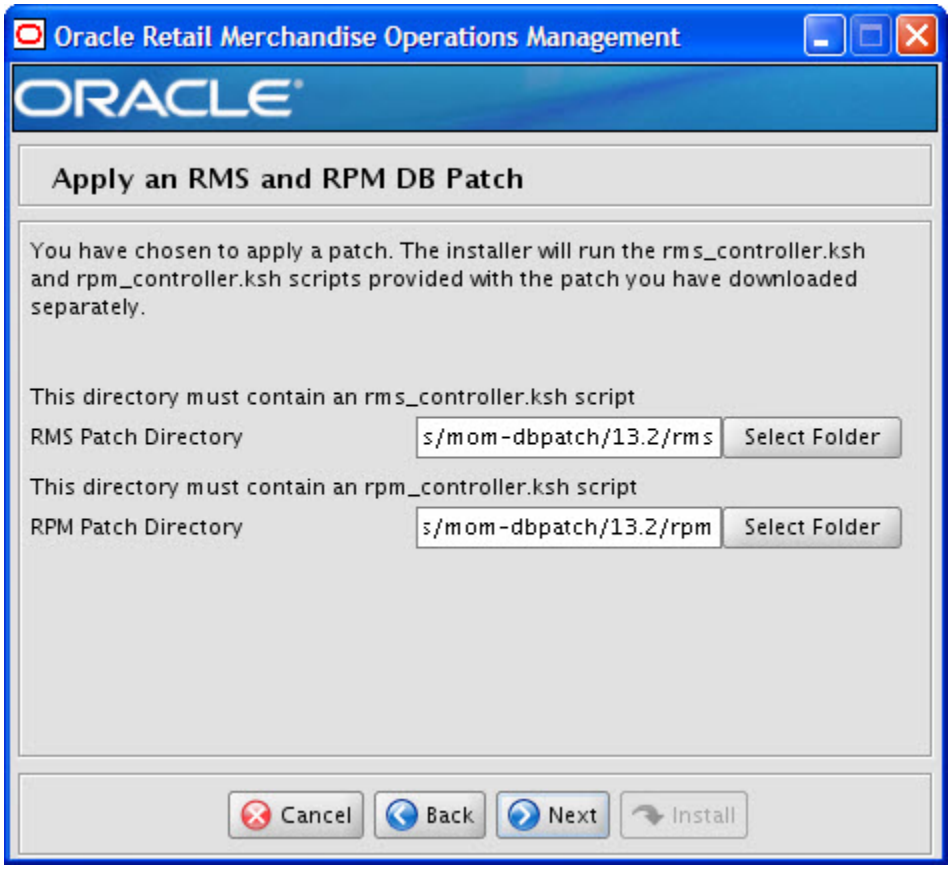


Field Title	DBA user
Field Description	Provide a database user with sufficient privileges to create synonyms between other users. The installer logs into the database using this account and create the synonyms needed between the RMS and Allocation users.
Example	SYSTEM

Field Title	DBA user password
Field Description	Database password for the DBA user.

The database settings provided are validated by the installer when you advance to the next screen.

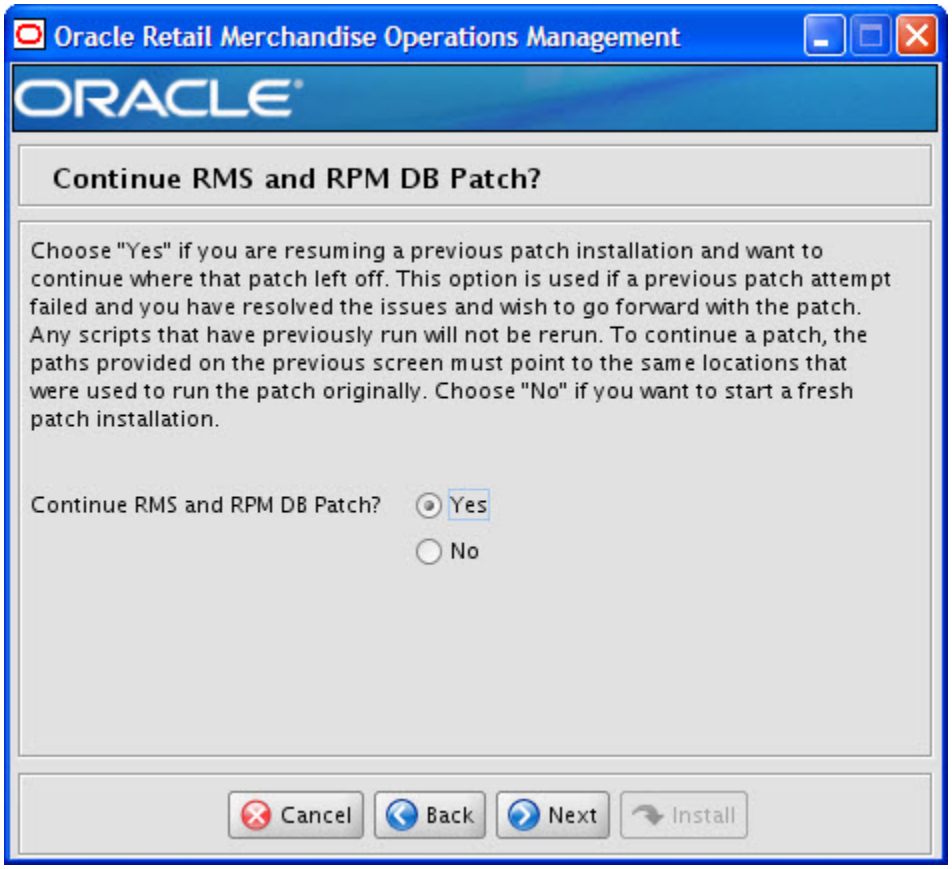
Screen: Apply an RMS and RPM DB Patch



Field Title	RMS Patch Directory
Field Description	This page appears if the Patch or Full+Patch option is selected on the earlier Full Or Patch screen. Provide the directory path to the downloaded RMS patch you want to install. The installer runs only the patch you provide. Note: The directory you choose must contain an rms_controller.ksh file.
Example	/my/rms/patch/dir for all 13.2.x patches Note: The patch option is intended for patches starting with 13.2.

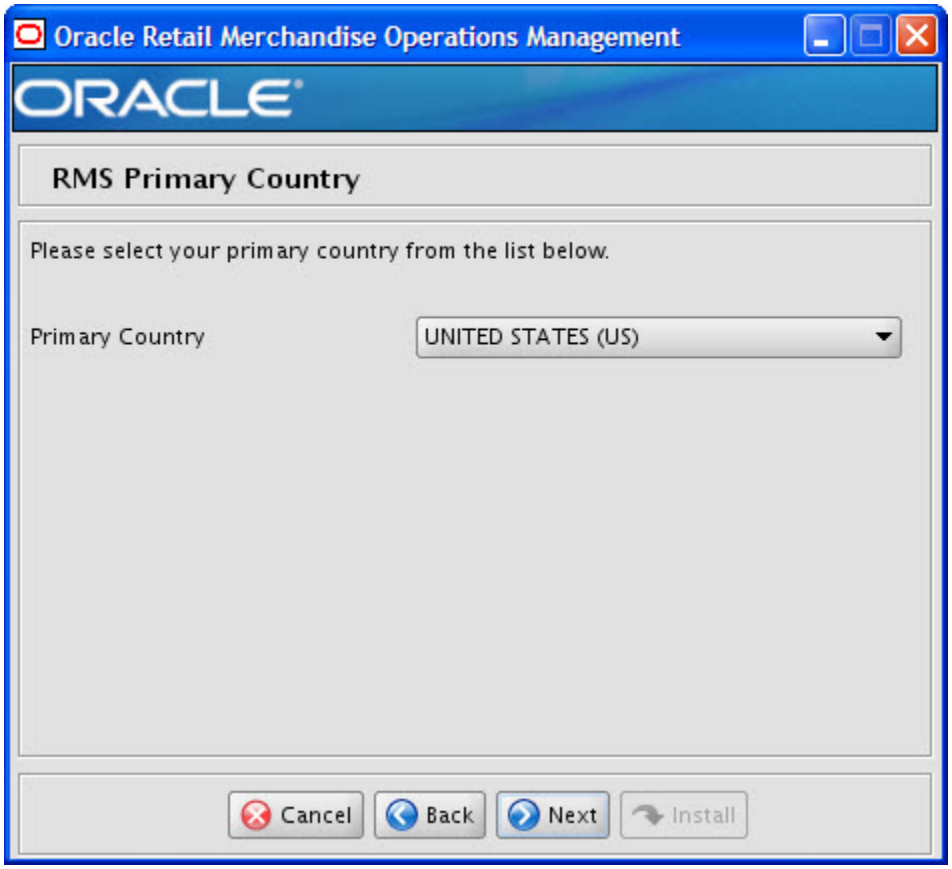
Field Title	RPM Patch Directory
Field Description	This page appears if the Patch or Full+Patch option is selected on the earlier Full Or Patch screen. Provide the directory path to the downloaded RPM patch you want to install. The installer runs only the patch you provide. Note: The directory you choose must contain an rpm_controller.ksh file.
Example	/my/rpm/patch/dir for all 13.2.x patches Note: The patch option is intended for patches starting with 13.2.

Screen: Continue RMS and RPM DB Patch



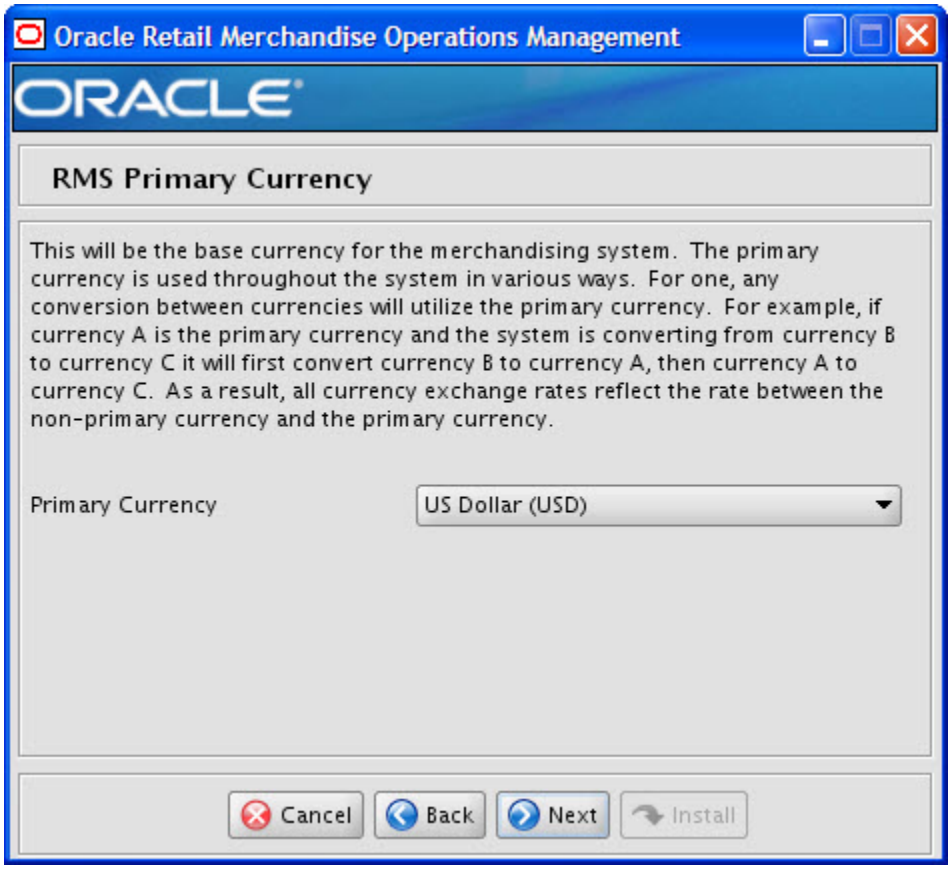
Field Title	Continue RMS and RPM DB Patch?
Field Description	The patch process allows you to continue a previously run patch if it stopped before completion or failed. If "Yes" is selected, any scripts that were previously run for the RMS and RPM patch will be skipped. If "No" is selected, the patch will start from the beginning. Note: To continue a patch, the content of the "processed" directories in the RMS Patch Directory and RPM Patch Directory 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 won't be able to continue this patch in the future.

Screen: RMS Primary Country



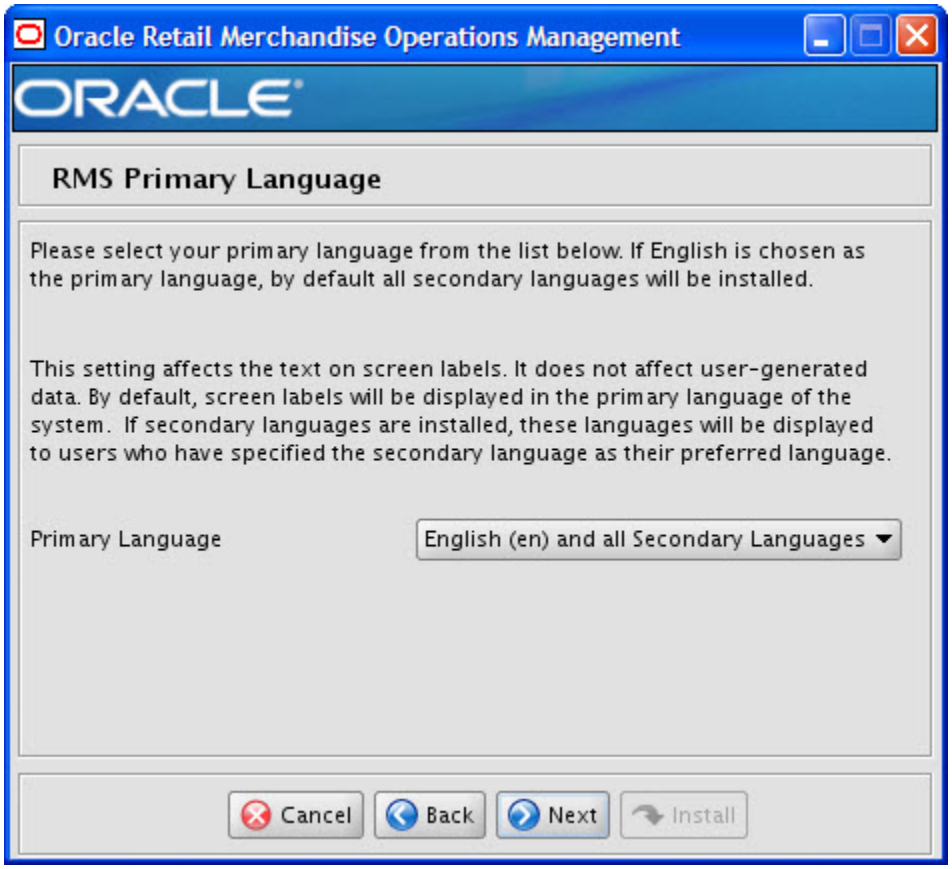
Field Title	Primary Country
Field Description	Choose your primary country from the list provided.
Example	UNITED STATES (US)

Screen: RMS Primary Currency



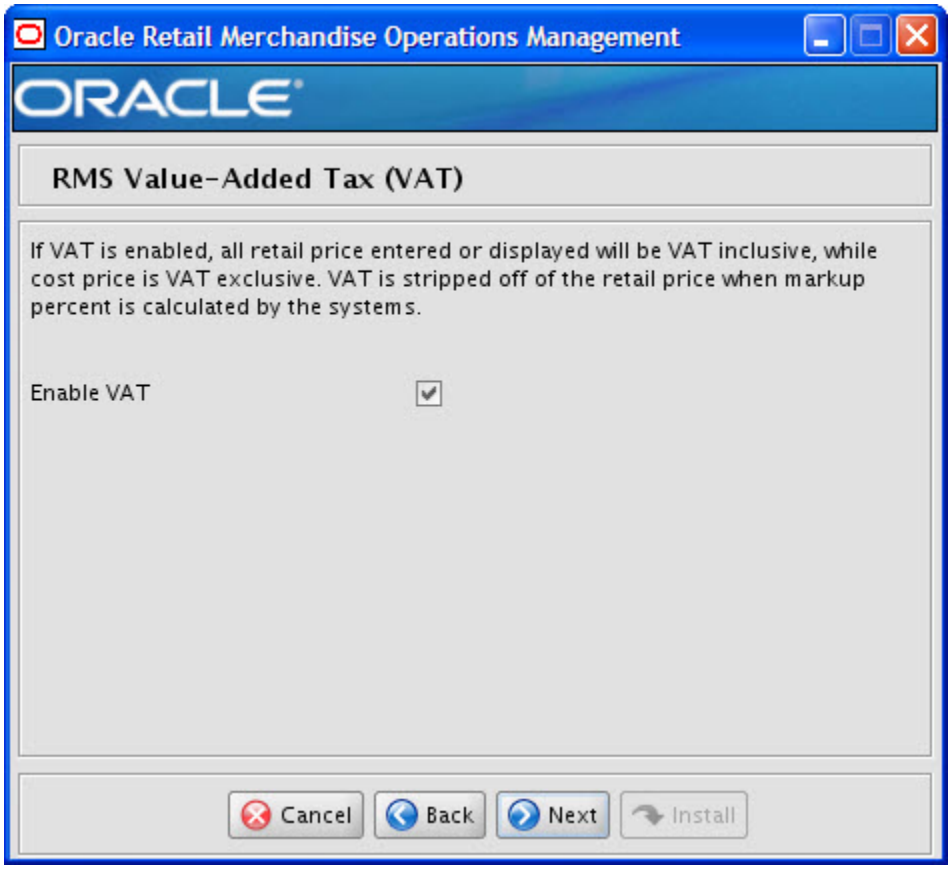
Field Title	Primary Currency
Field Description	Choose your primary currency from the list provided.
Example	US DOLLAR (USD)

Screen: RMS Primary Language



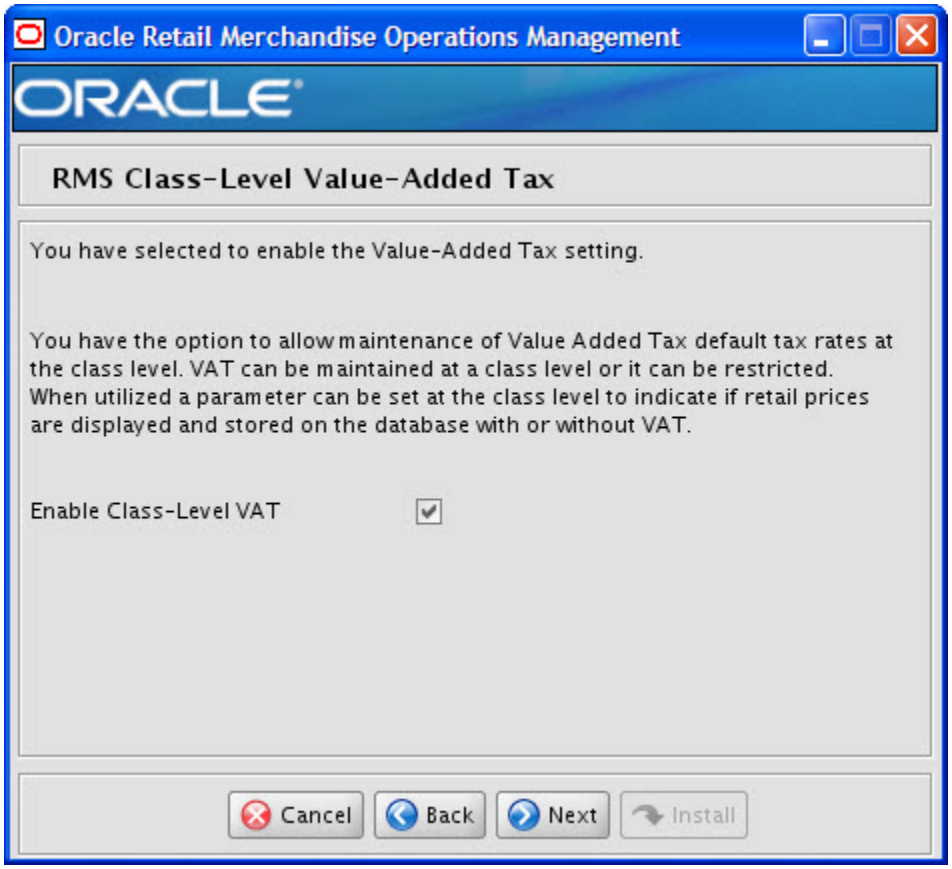
Field Title	Primary Language
Field Description	Choose your primary language from the list provided. By default, all secondary languages will be installed. If you choose a non-English primary language, no secondary languages will be installed.
Example	English (en) and all Secondary Languages

Screen: RMS Value-Added Tax (VAT)



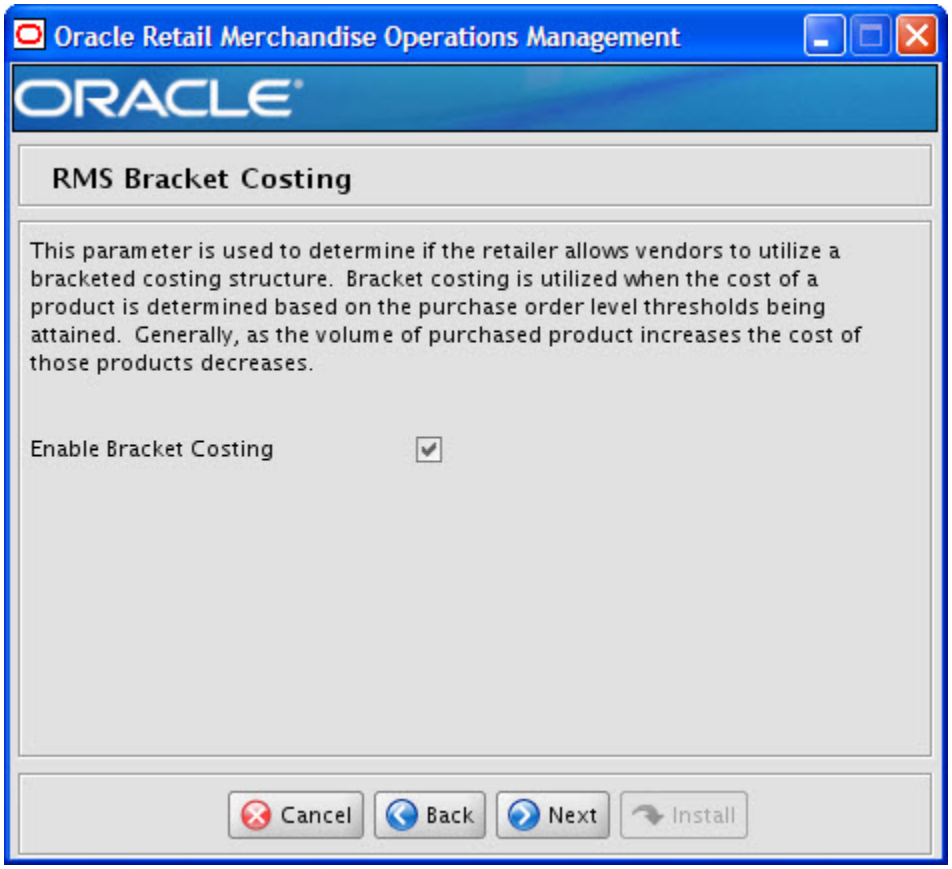
Field Title	Enable VAT?
Field Description	Select Yes if you will use VAT.

Screen: RMS Class Level Value-Added Tax



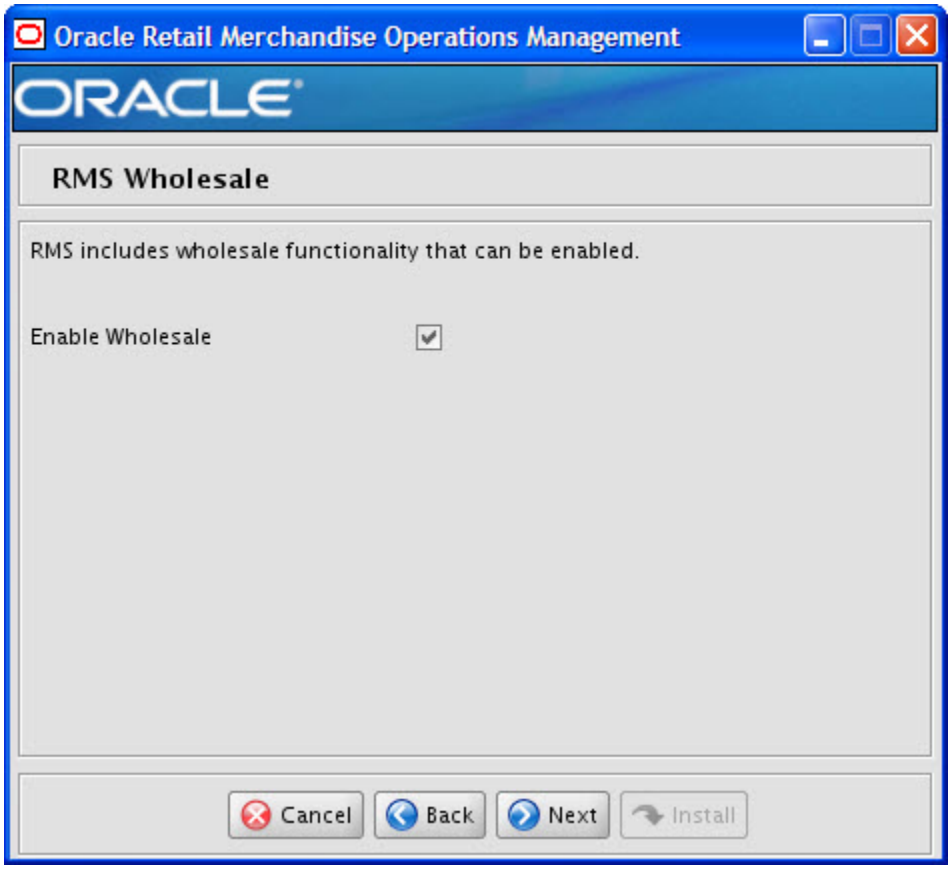
Field Title	Enable Class-Level VAT?
Field Description	You said yes to the VAT setting. Select Yes in this field to allow tax rates to be maintained at the class level. Select No to restrict tax rates.

Screen: RMS Bracket Costing



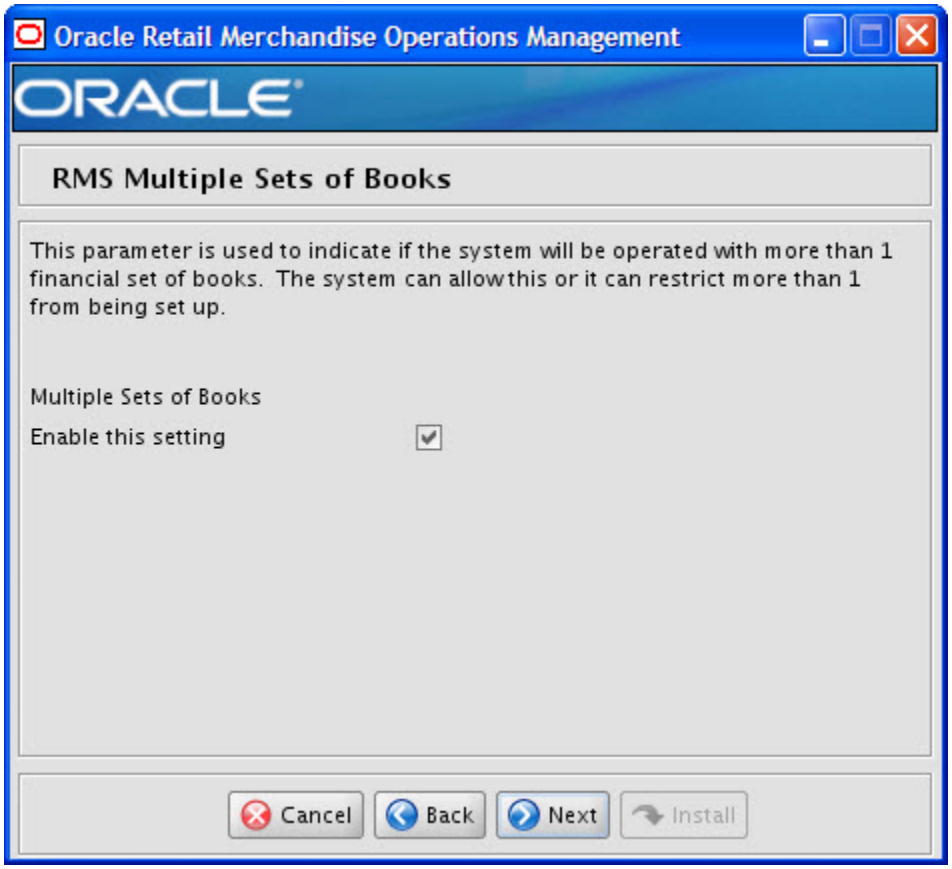
Field Title	Enable Bracket Costing?
Field Description	Select Yes if you allow vendors to use a bracketed costing structure.

Screen: RMS Wholesale



Field Title	Enable Wholesale?
Field Description	Wholesale functionality in RMS

Screen: RMS Multiple Sets of Books

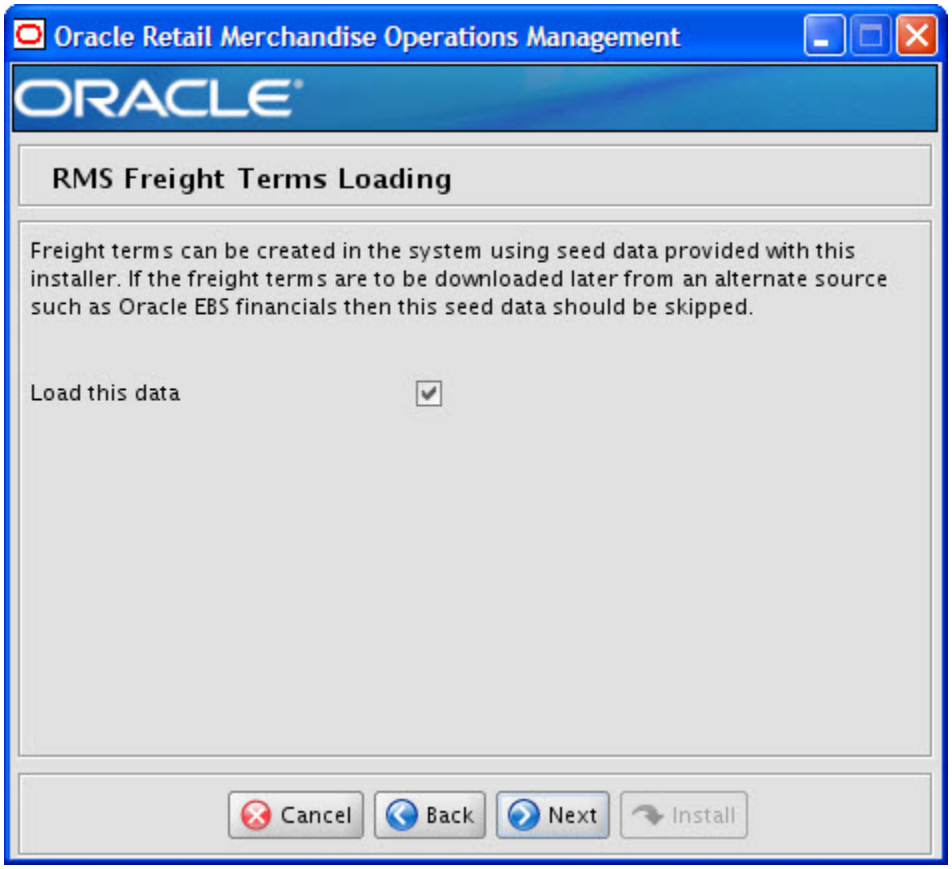


Field Title	Enable Multiple Sets of Books?
Field Description	Indicates whether the system will be operated with more than one financial set of books.

Screen: RMS Supplier Sites

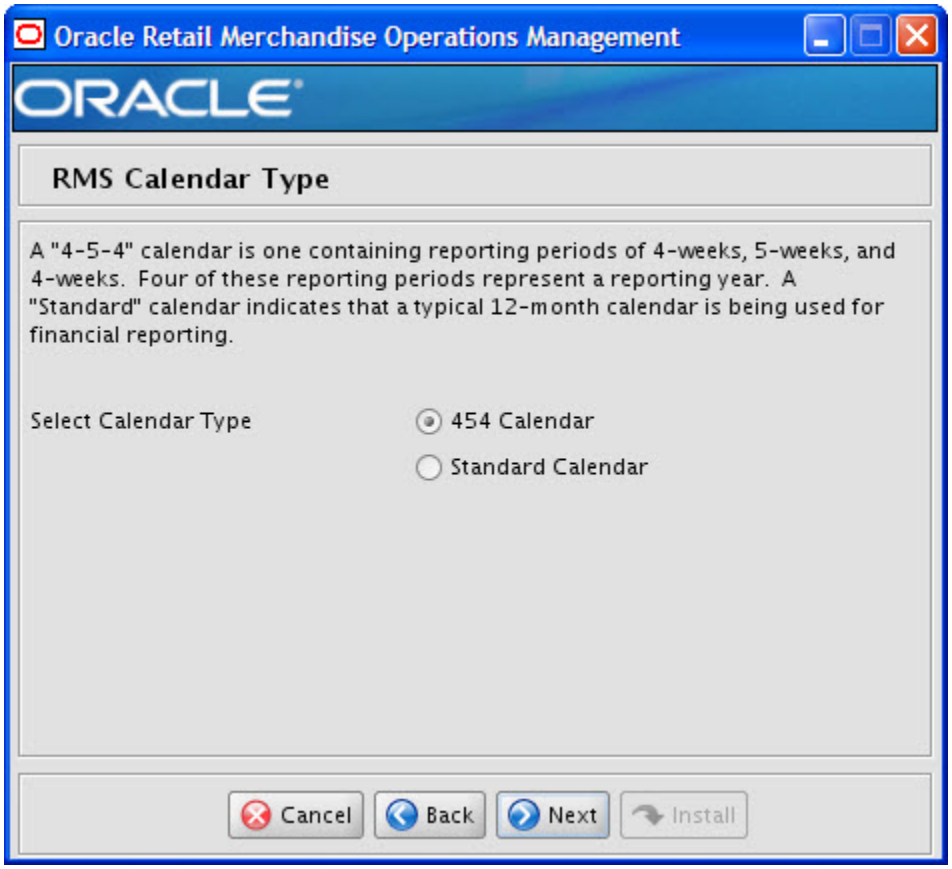
Field Title	Enable Supplier Sites?
Field Description	Indicates if a supplier hierarchy is being utilized in the system.

Screen: RMS Freight Terms Loading



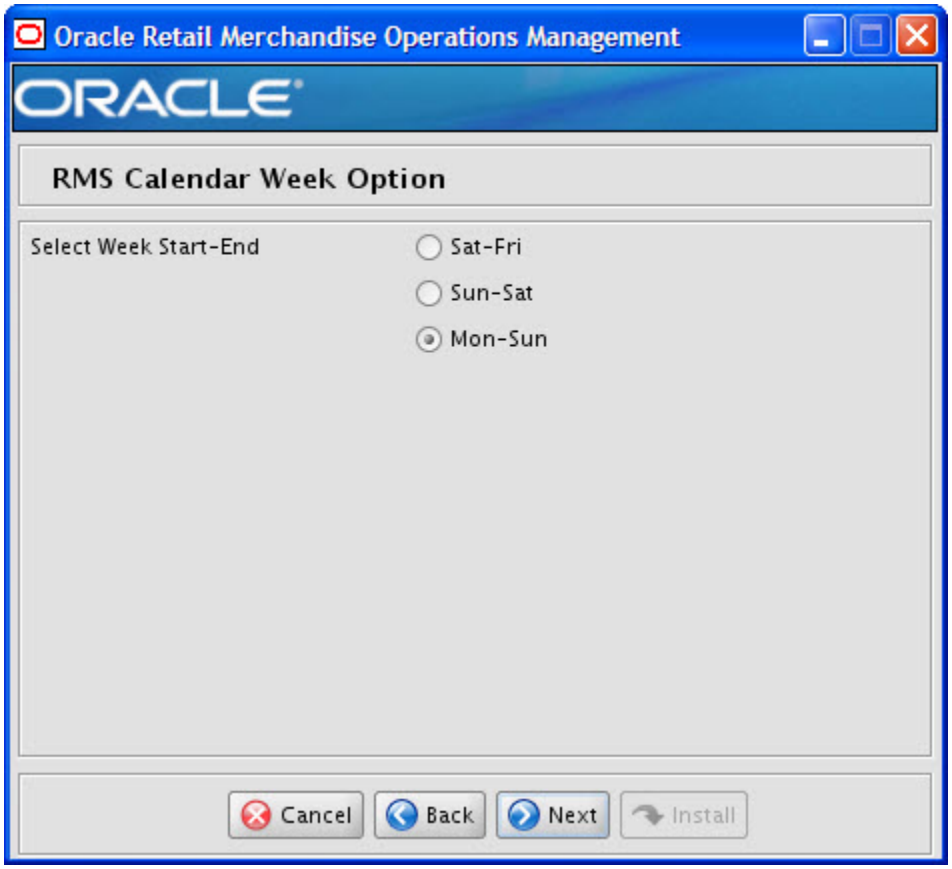
Field Title	Load this data
Field Description	Freight terms data is provided with the RMS release. Select this option to insert it into the schema. If data will be pulled from another system such as EBS financials then do not select this option.

Screen: RMS Calendar Type



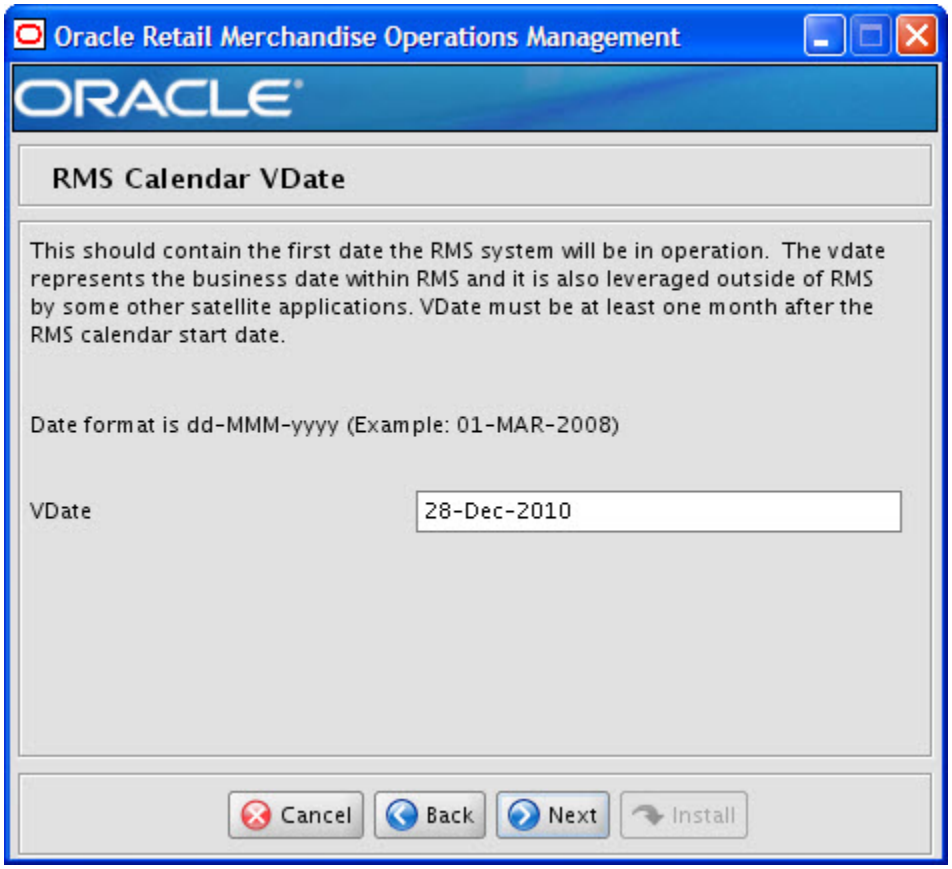
Field Title	Calendar Type
Example	454 Calendar

Screen: RMS Calendar Week Option



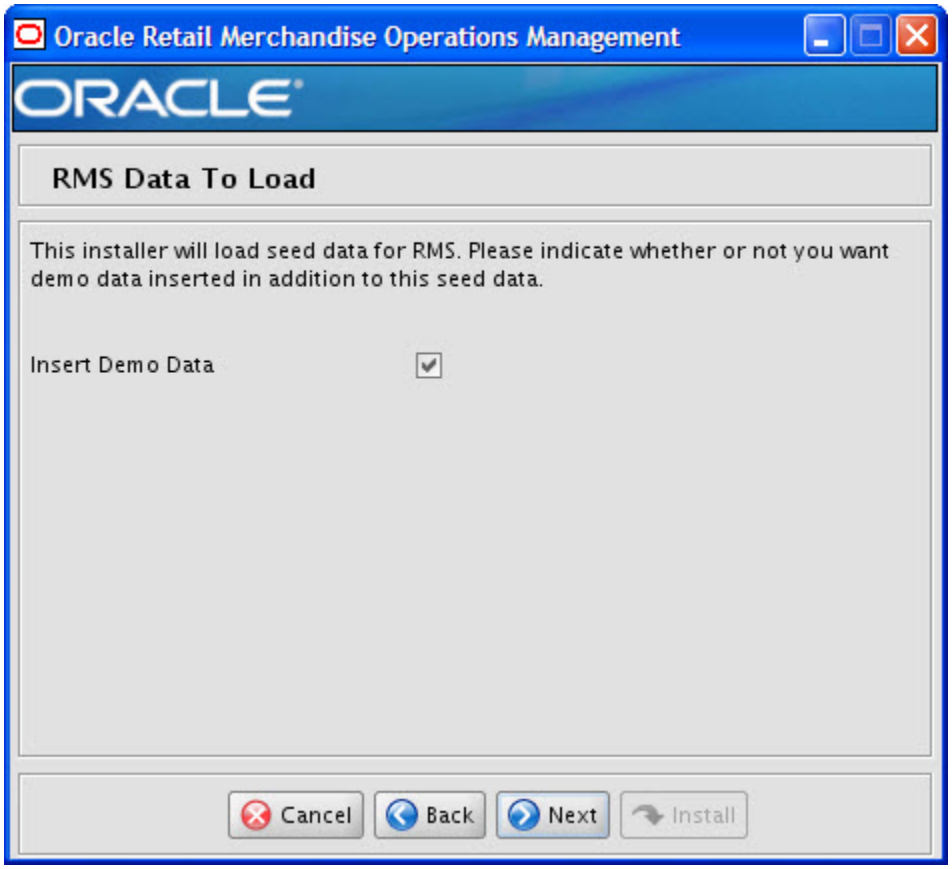
Field Title	Week Start-End
Example	Mon-Sun

Screen: RMS Calendar VDate



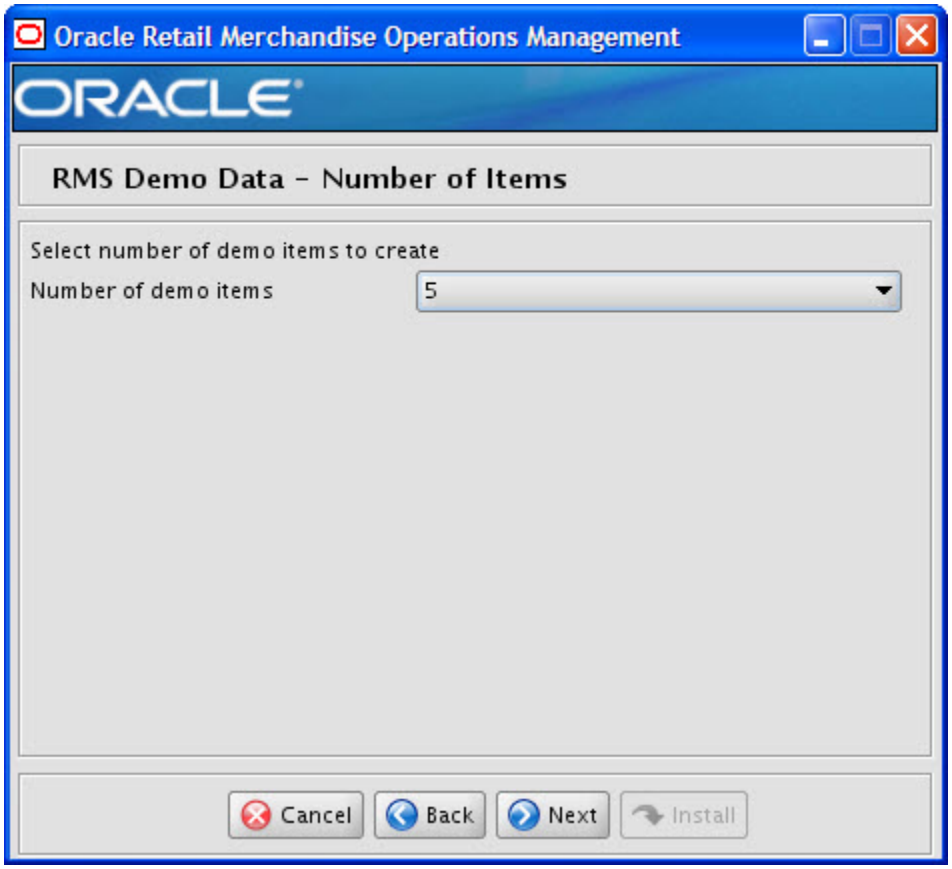
Field Title	VDate
Example	28-Dec-2010

Screen: RMS Data to Load



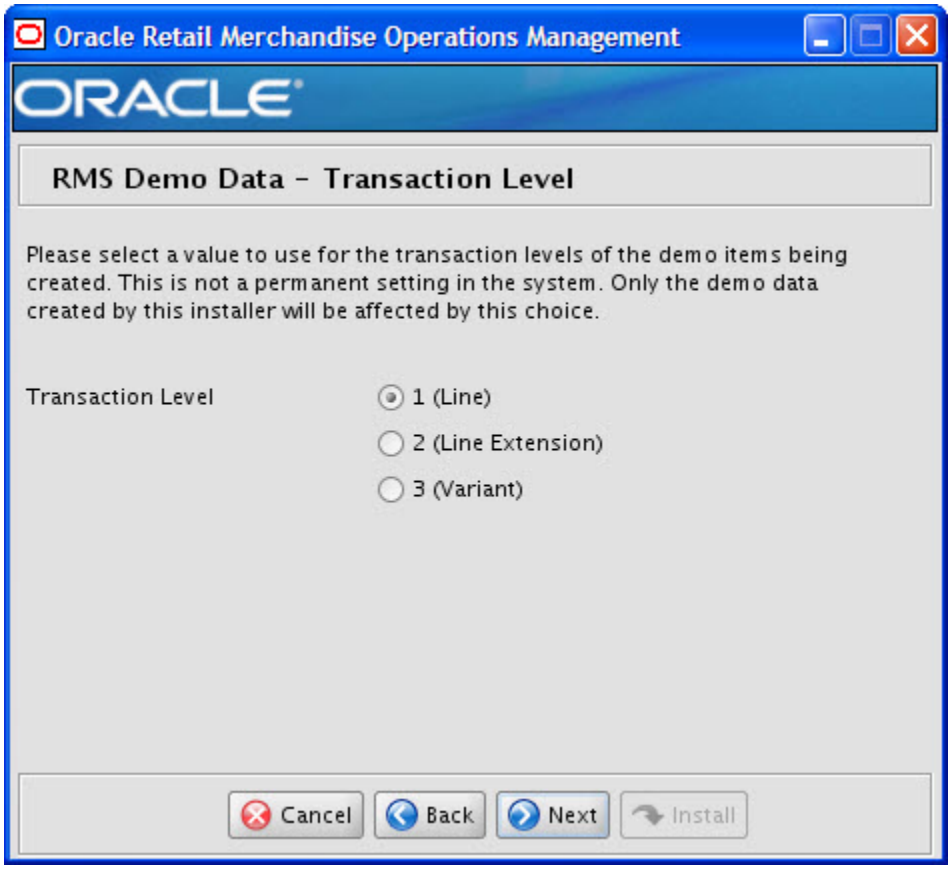
Field Title	Insert demo data
Field Description	Indicate whether or not the demo data scripts should be run for RMS.

Screen: RMS Demo Data – Number of Items



Field Title	Number of demo items
Field Description	If you chose to insert demo data, this setting determines how many demo items to create.
Example	5

Screen: RMS Demo Data - Transaction Level

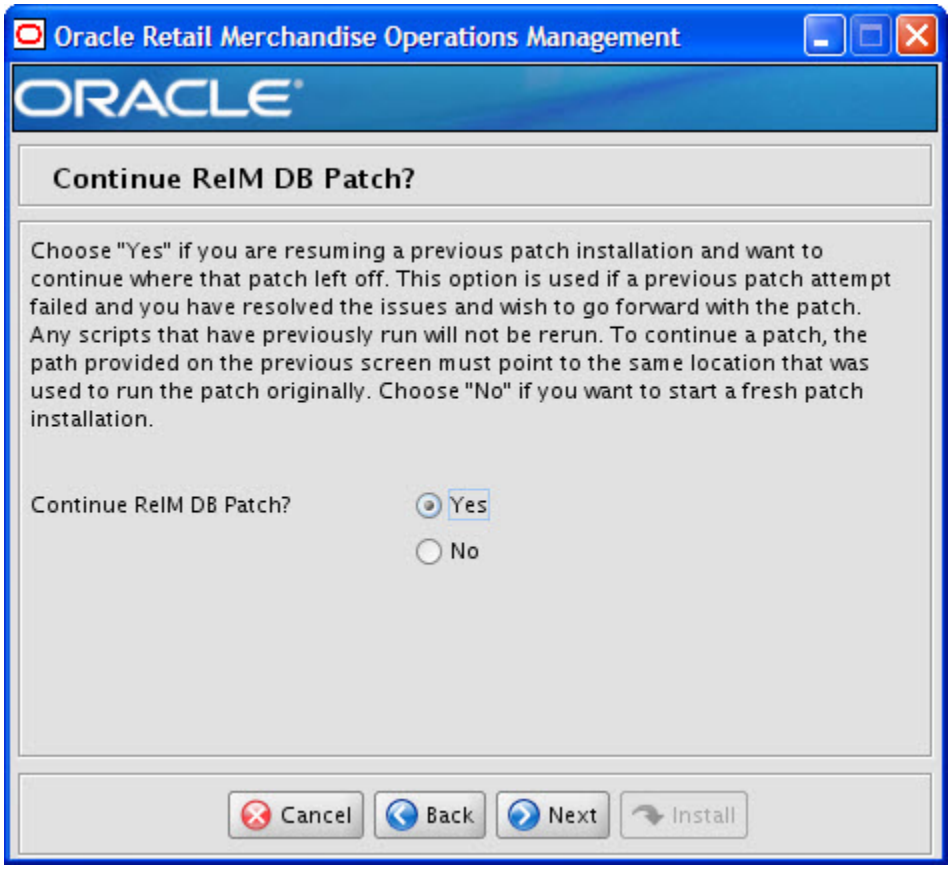


Field Title	Transaction Level
Field Description	If you chose to insert demo items on the previous screen, you are asked to provide a transaction level value for these items.
Example	1

Screen: Apply RelM DB Patch

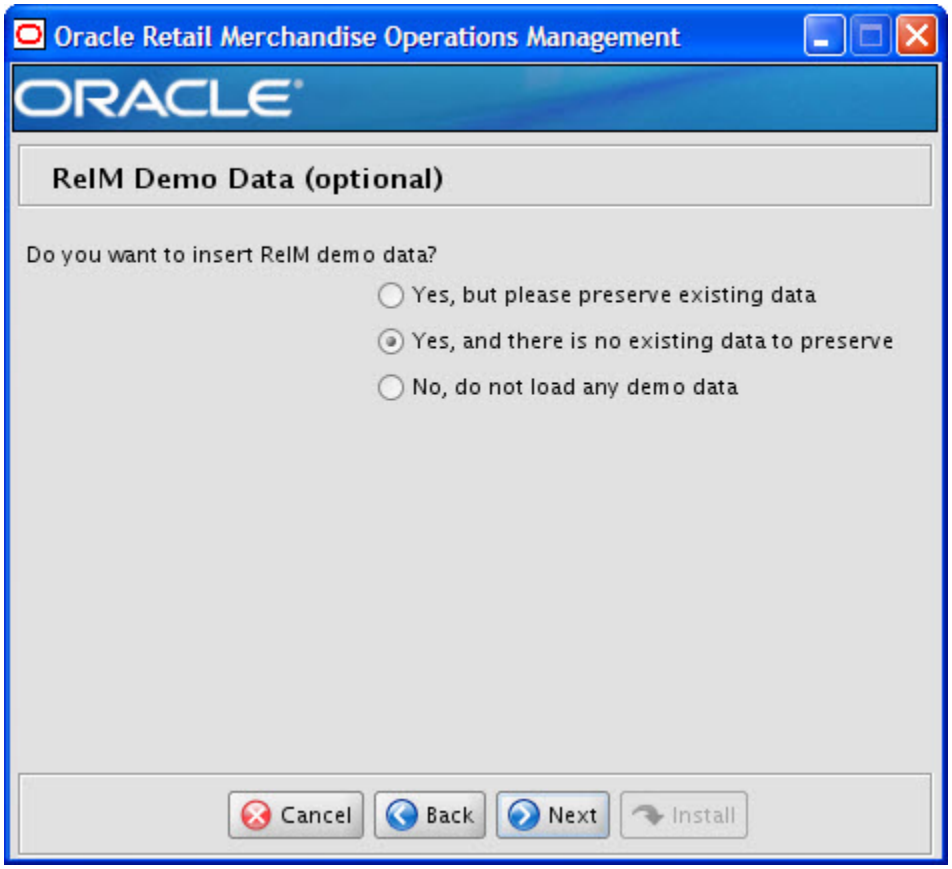
Field Title	Patch Directory
Field Description	This page appears if the Patch or Full+Patch option is selected on the earlier Full Or Patch screen. Provide the directory path to the downloaded ReIM patch you want to install. The installer runs only the patch you provide. Note: The directory you choose must contain an reim_controller.ksh file.
Example	/my/reim/patch/dir for all 13.2.x patches Note: The patch option is intended for patches starting with 13.2.

Screen: Continue ReIM DB Patch



Field Title	Continue ReIM 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 ReIM 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" directory in the Patch Directory 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 won't be able to continue this patch in the future.

Screen: ReIM Demo Data (optional)



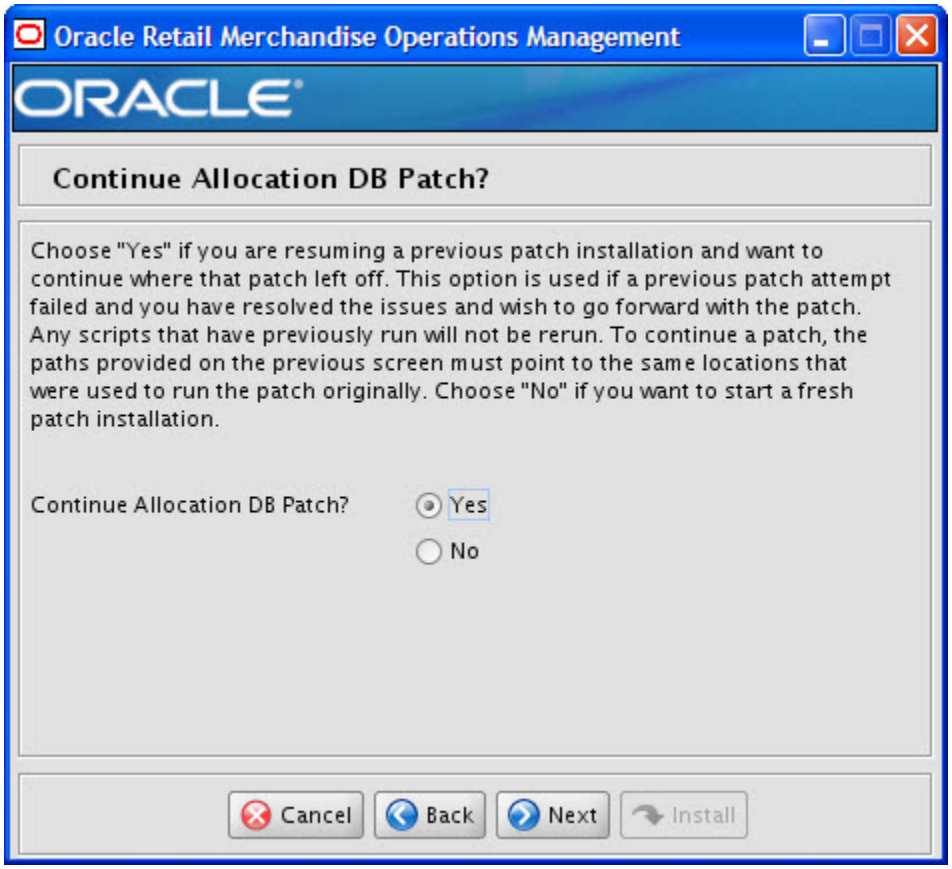
Field Title	Do you want demo data inserted into the RMS TERMS and INVC* tables?
Field Description	Select "Yes, and there is no existing data to preserve" for ReIM demo data to be inserted into the ReIM/RMS schema. Note: ReIM demo data must be run to insert sample ReIM users for logging into the ReIM application.

Screen: Apply Allocation DB Patch

Field Title	Alloc Patch Directory
Field Description	This page appears if the Patch or Full+Patch option is selected on the earlier Full Or Patch screen. Provide the directory path to the downloaded patch for the Allocation schema you want to install. The installer runs only the patch you provide. Note: The directory you choose must contain an alloc_controller.ksh file.
Example	/my/alloc/patch/dir for all 13.2.x patches Note: The patch option is intended for patches starting with 13.2.

Field Title	Alloc in RMS Patch Directory
Field Description	This page appears if the Patch or Full+Patch option is selected on the earlier Full Or Patch screen. Provide the directory path to the downloaded Allocation patch for the RMS Schema you want to install. The installer runs only the patch you provide. Note: The directory you choose must contain an alloc_rms_controller.ksh file.
Example	/my/alloc_rms/patch/dir for all 13.2.x patches Note: The patch option is intended for patches starting with 13.2.

Screen: Continue Allocation DB Patch



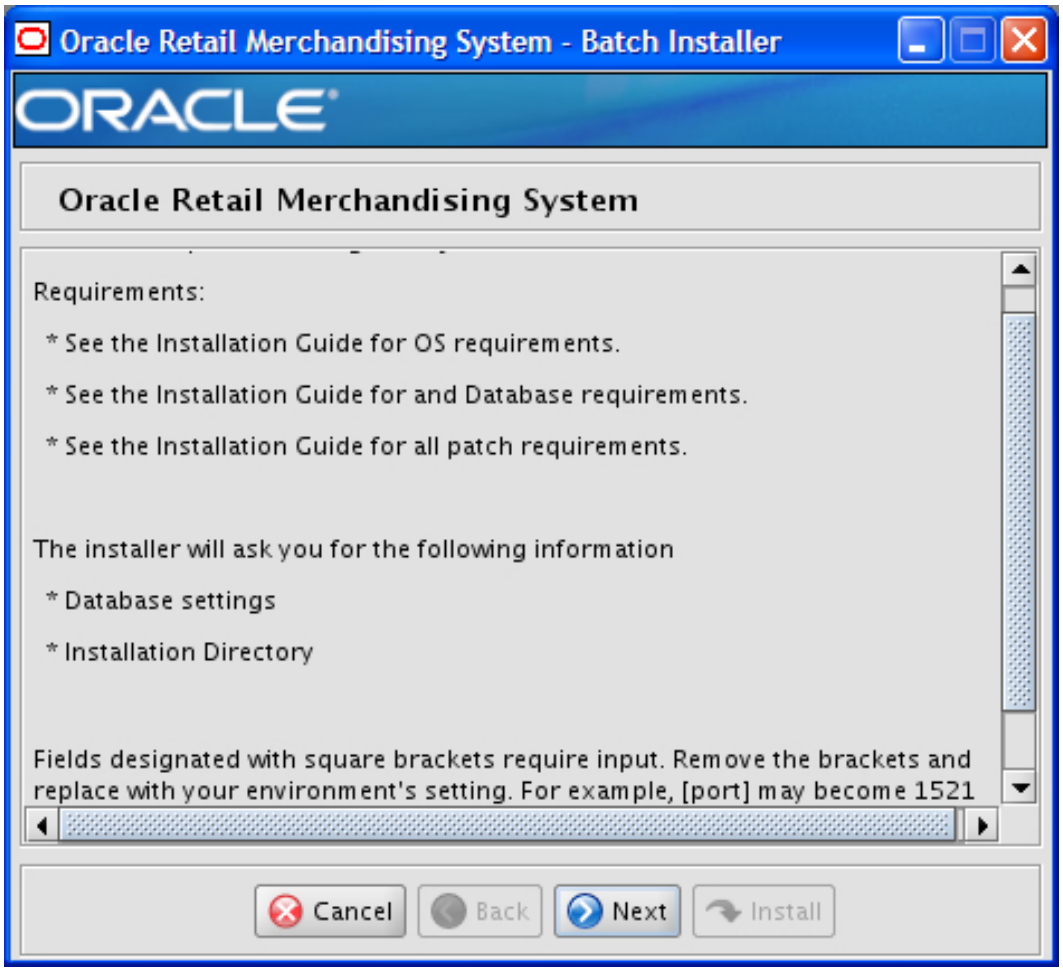
Field Title	Continue Allocation DB Patch?
Field Description	The patch process allows you to continue a previously run patch if it stopped before completion or failed. If Yes is selected, any scripts that were previously run for the Allocation patch in the Allocation and RMS schemas will be skipped. If No is selected, the patch will start from the beginning. **Note**: To continue a patch, the content of the processed directories in the Alloc Patch Directory and Alloc in RMS Patch Directory 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 won't be able to continue this patch in the future.

H

Appendix: RMS Batch Installer Screens

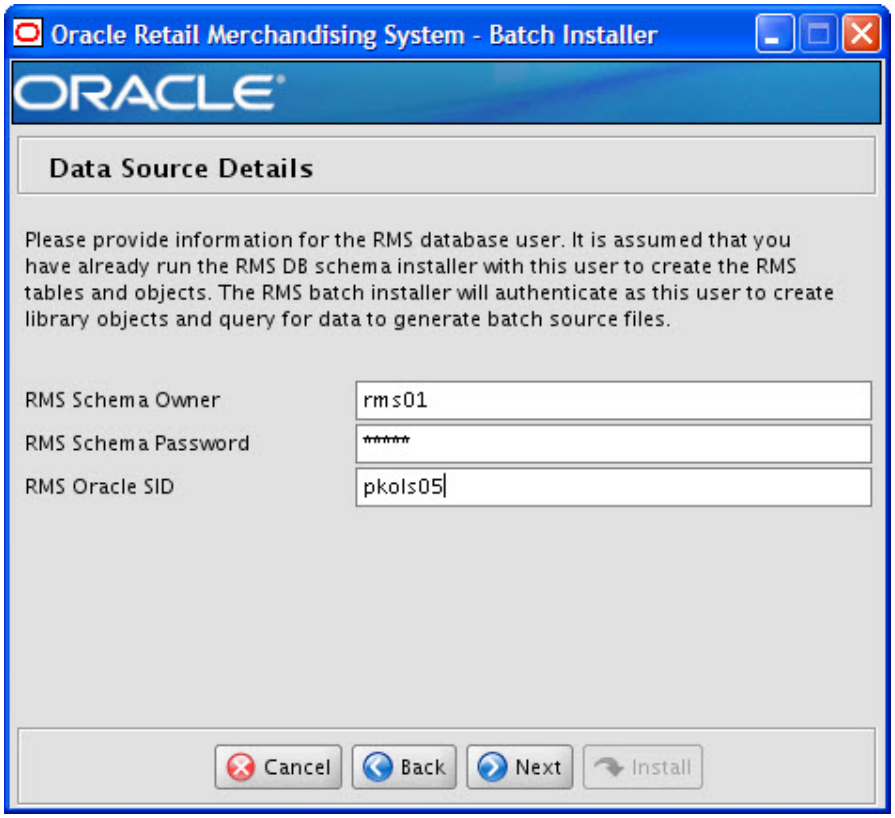
You need the following details about your environment for the installer to successfully compile and install the RMS batch programs. Depending on the options you select, you may not see some screens or fields.

Screen: Welcome



There are no fields on this screen. The Welcome screen contains information about the RMS Batch Installer and prerequisites.

Screen: DataSourceDetails

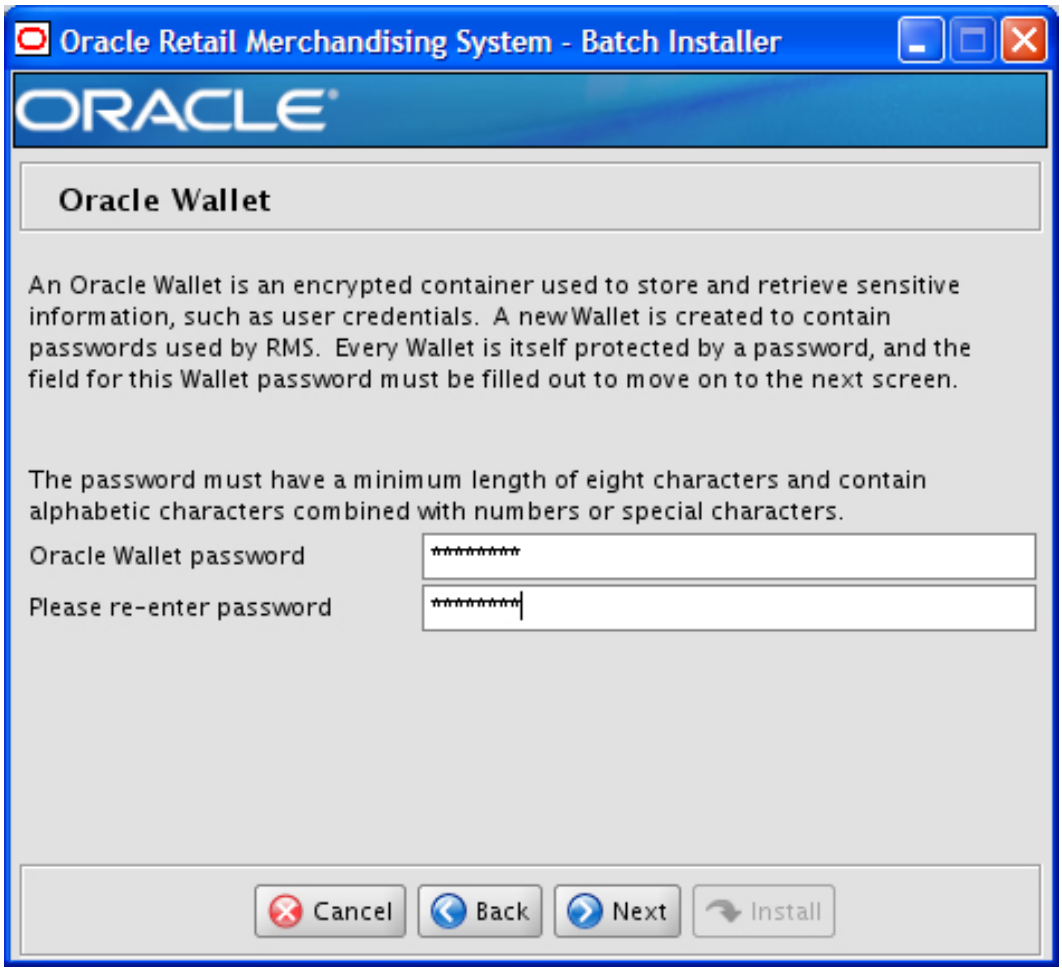


Field Title	RMS Schema Owner
Field Description	Provide the RMS database user here. The installer will log into the database as this user to create RMS library objects and query for data to generate batch source files. This user must already exist in the database and have the RMS tables installed.
Example	rms01

Field Title	RMS Schema Password
Field Description	Database password for the RMS Schema Owner.

Field Title	RMS Oracle SID
Field Description	Oracle system identifier for the database where RMS will be installed
Example	pkols05

Screen: Oracle Wallet

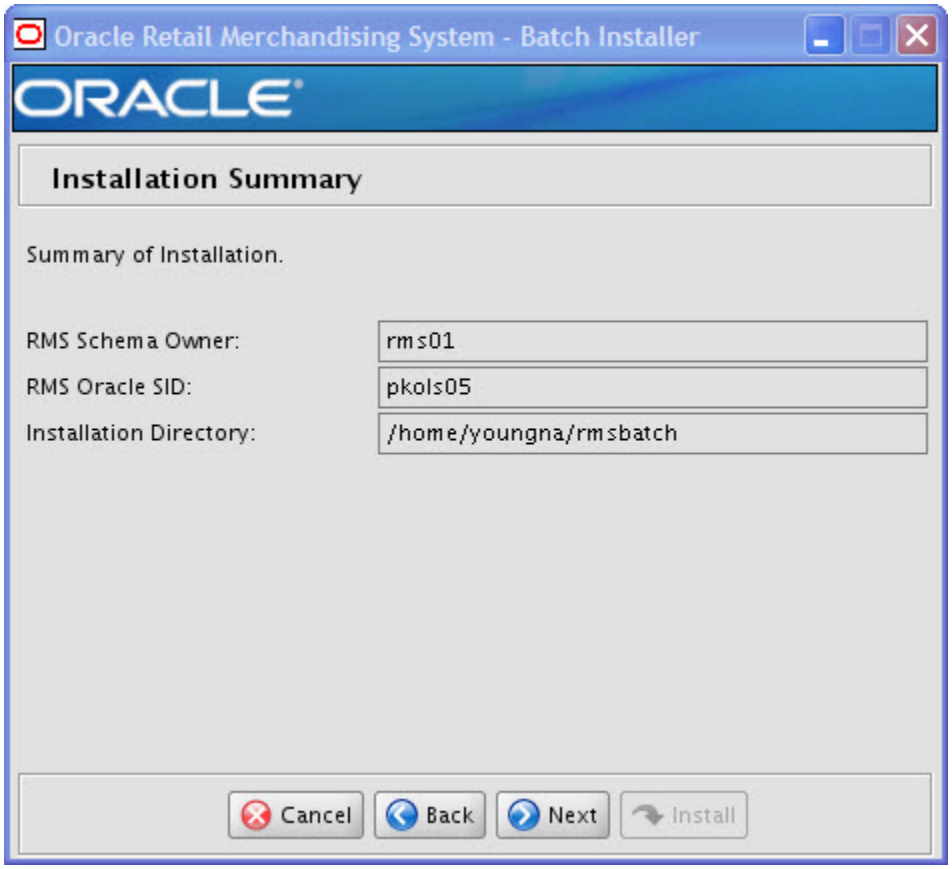


Field Title	Oracle Wallet password
Field Description	This is the password for the wallet that will store the database credentials that were supplied in the previous screen.

Screen: Batch Installation Directory

Field Title	Batch Installation Directory
Field Description	Location where the installer will install the batch source and then compile it. This is the permanent location for the RMS batch programs.
Example	/opt/oracle/retail/rmsbatch

Screen: Summary



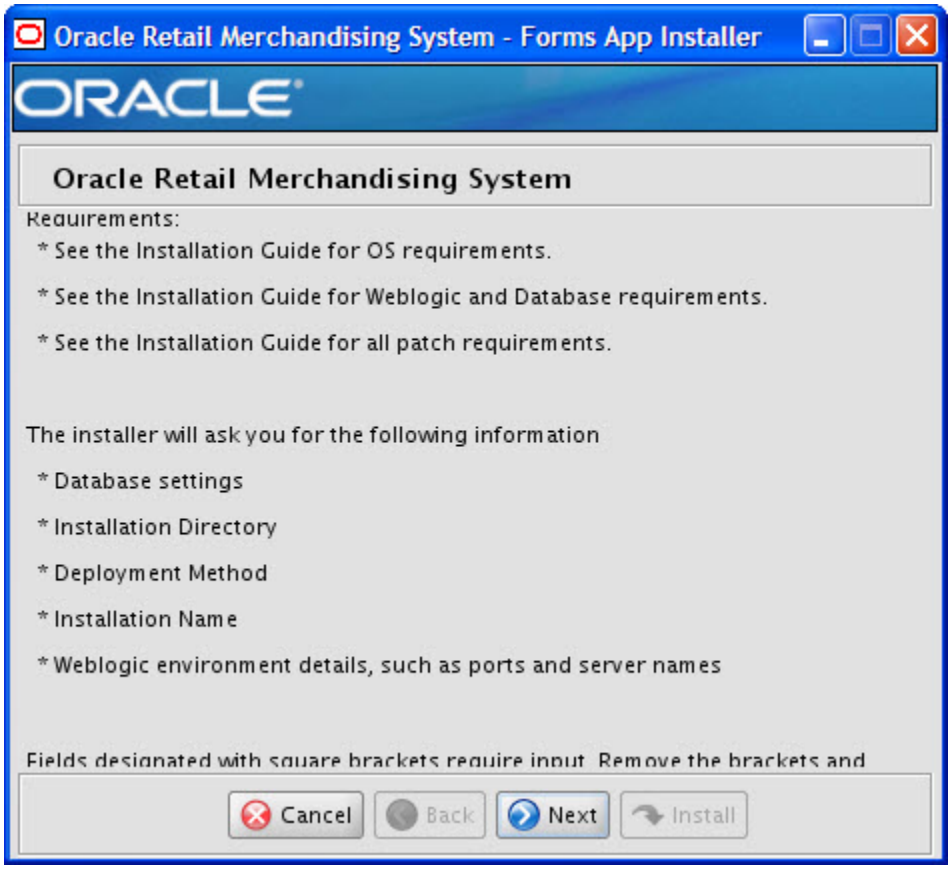
All of the fields on this summary screen are read-only. In GUI mode of the installer, this screen provides the opportunity to review inputs and go back to previous screens to correct them if necessary.

Once you advance forward from this screen, the installer connects to the database and validate that the RMS user exists before beginning installation.

I

Appendix: RMS Application Installer Screens

Screen: Welcome

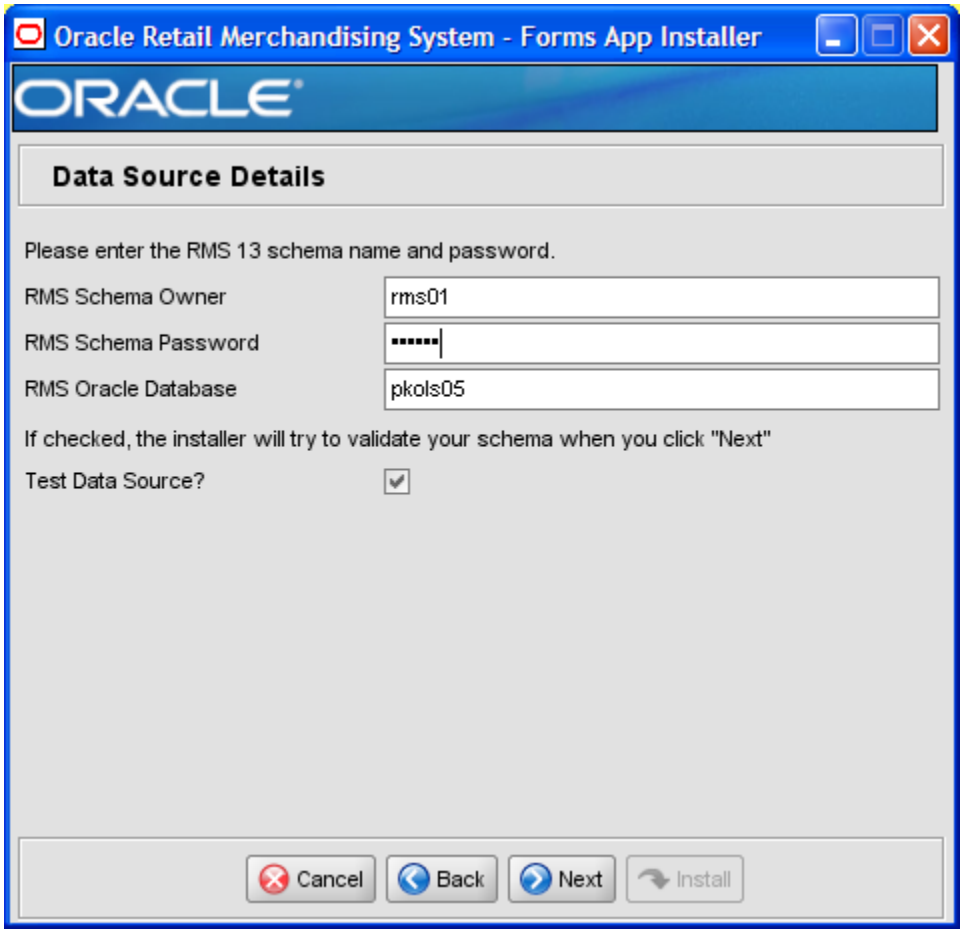


There are no fields on this screen. The Welcome screen contains information about the RMS Application Installer and prerequisites.

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



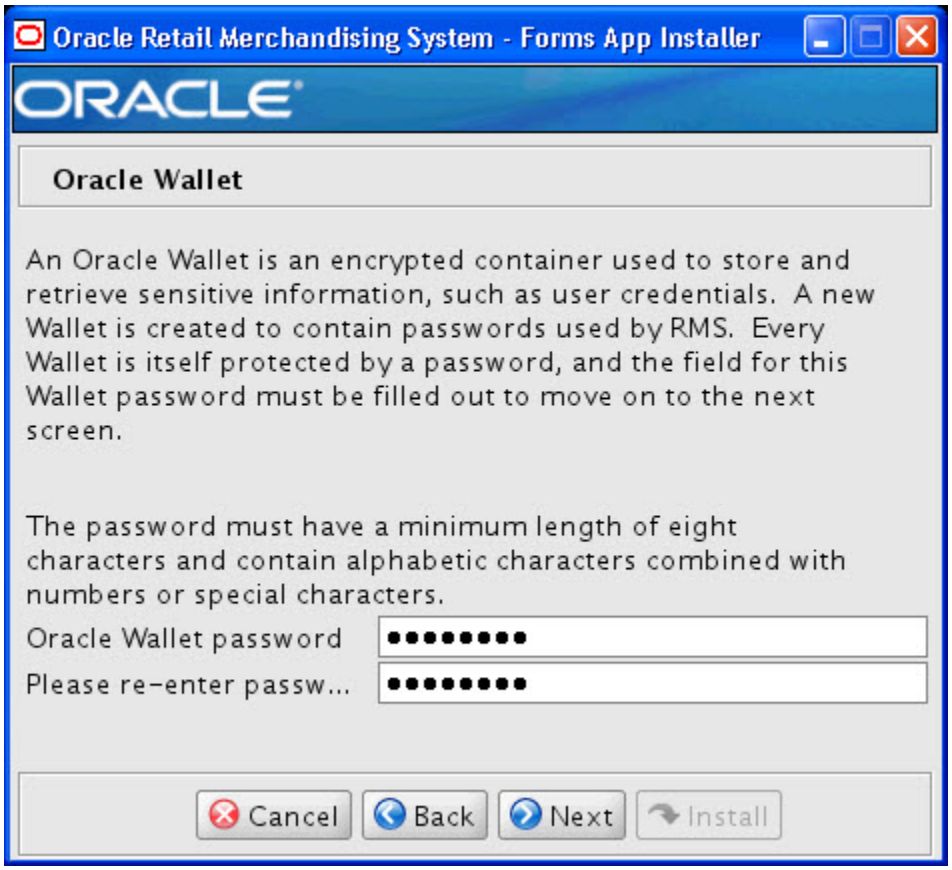
Field Title	RMS Schema Owner
Field Description	This is the same username that was used during the RMS Database Schema Installer.
Example	rms01

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

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

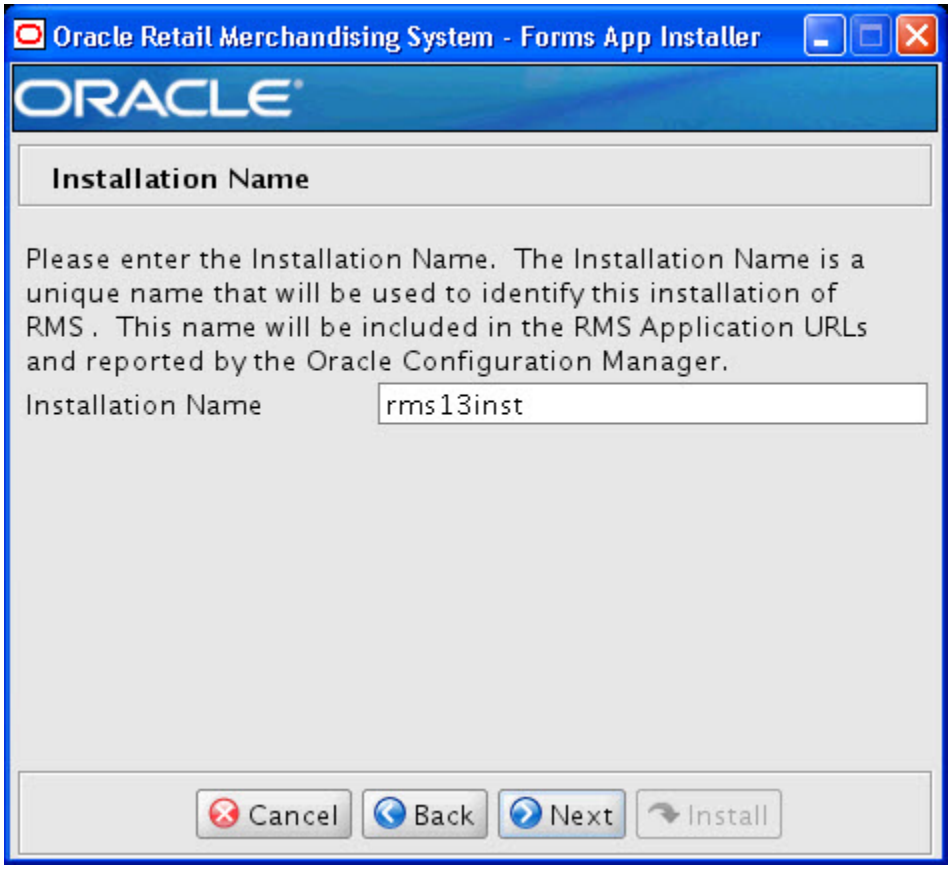
Field Title	Test Data Source?
Field Description	Attempt to validate the Data Source Details on this screen. This will happen when you click "Next" **Note:** If you get any errors not related to incorrectly entered credentials, please refer to Appendix L: Common Installation Errors.

Screen: Oracle Wallet



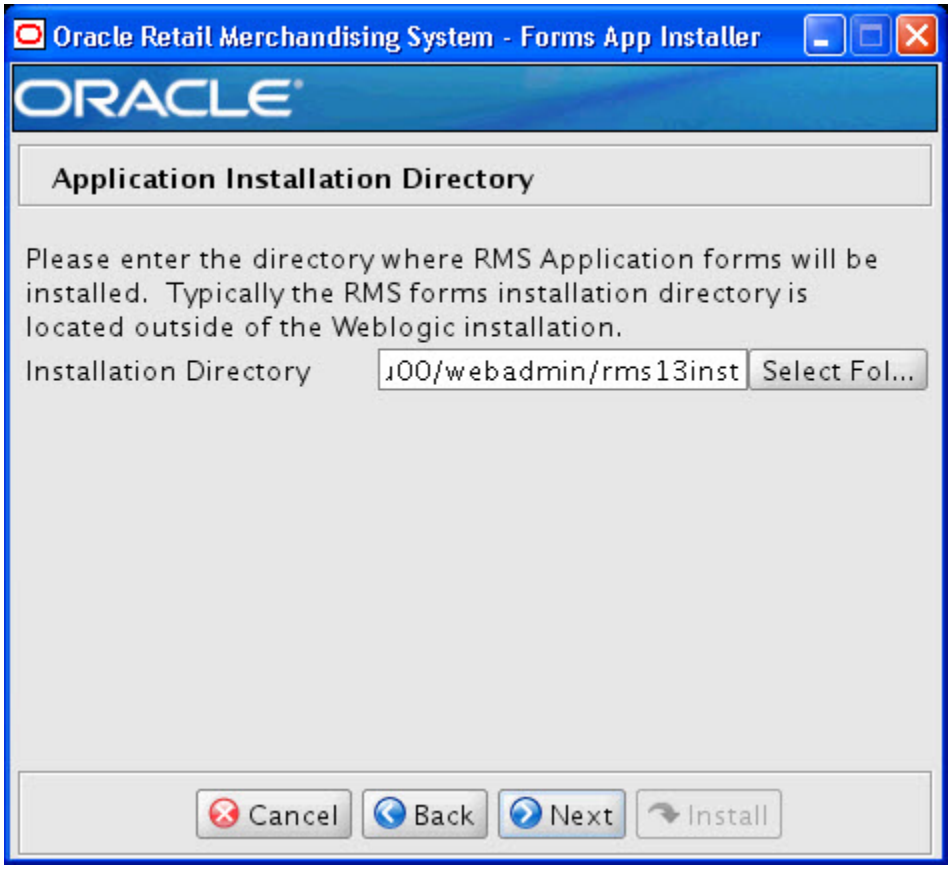
Field Title	Oracle Wallet password
Field Description	This is the password for the wallet that will store the database credentials that were supplied in the previous screen.

Screen: Installation Name



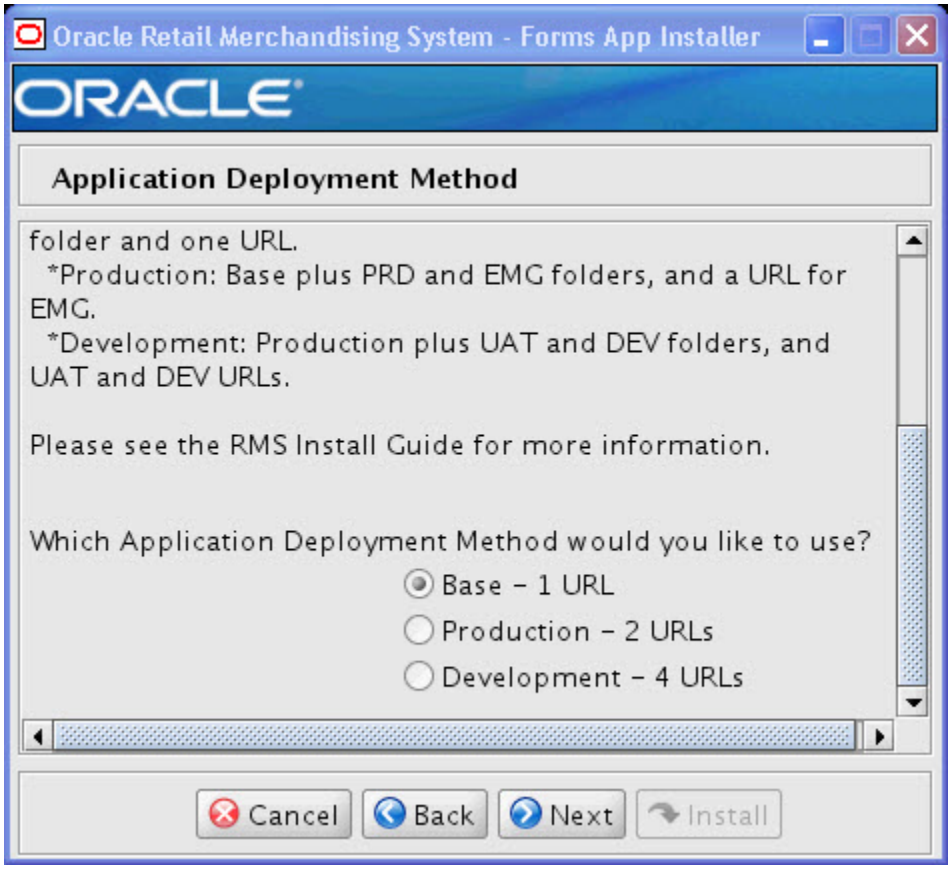
Field Title	Installation Name
Field Description	This value is used in conjunction the Oracle Configuration Manager (OCM). It gives the installation a unique name so the OCM can identify different installations of RMS in the same WebLogic instance.
Example	rms13inst

Screen: Application Installation Directory



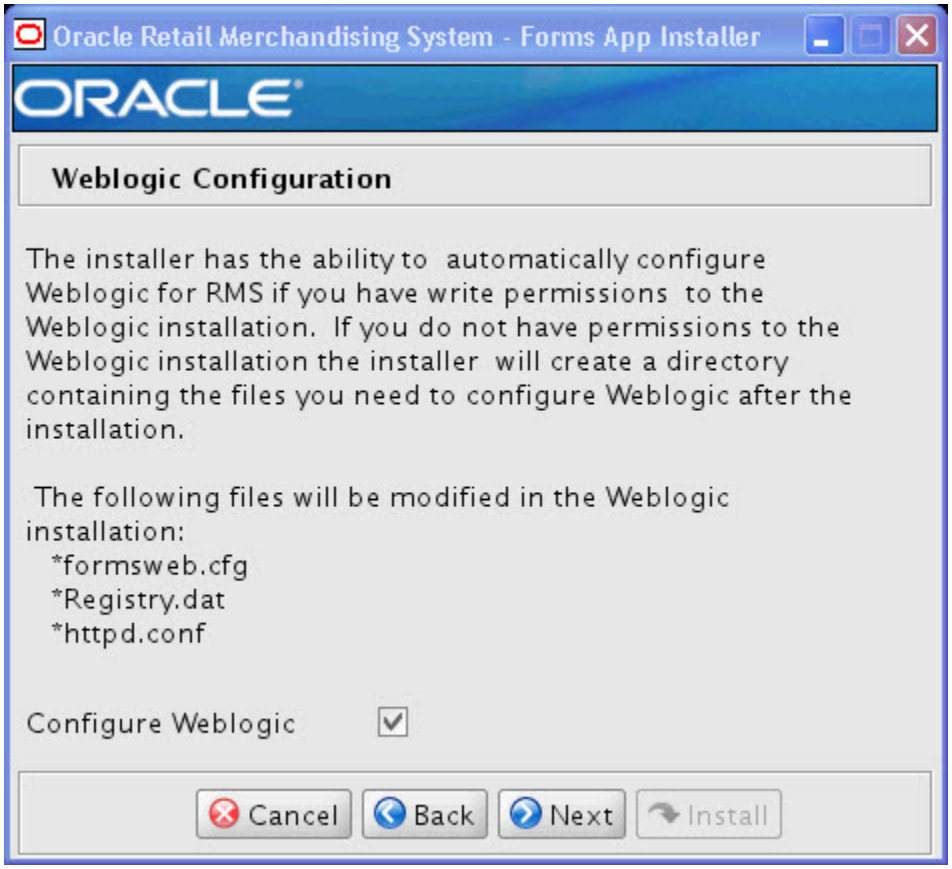
Field Title	Application Installation Directory
Field Description	The location where the RMS Application (toolset, forms and reports) will be installed. The RMS $MMHOME path will be a subdirectory of this directory, named base.
Example	/u01/oracle/retail

Screen: Application Deployment Method



Field Title	Which Environment Deployment Method would you like to use
Field Description	Select the Application Deployment Method you would like. Reference Appendix O for more information.
Example	Base

Screen: WebLogic Configuration



Field Title	Configure WebLogic
Field Description	Make the necessary configurations to the WebLogic server to be able to run RMS 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 cleanup duplicate entries in the WebLogic formsweb.cfg file.

Screen: Weblogic Administrative Details

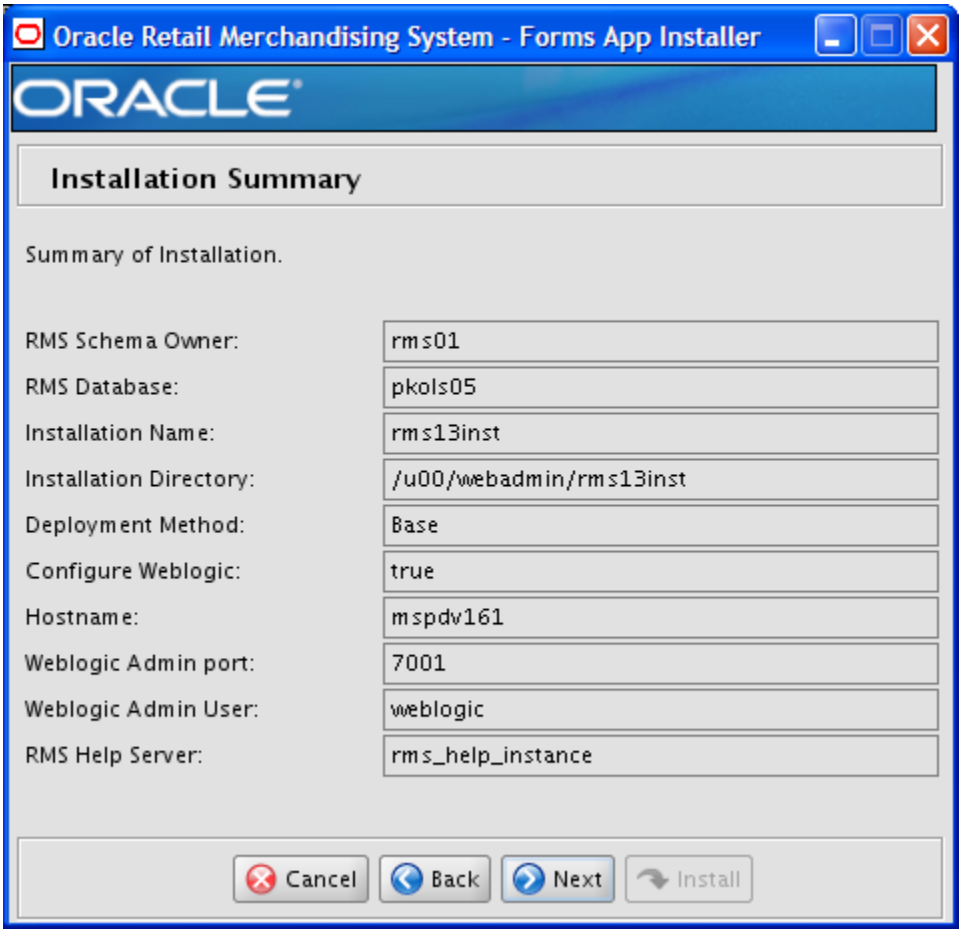
Field Title	WebLogic Admin User
Field Description	Username of the admin user for WebLogic instance to which the RMS Webhelp application is being deployed.
Example	weblogic

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

Screen: Webhelp Installation Details

Field Title	WebLogic Help Server
Field Description	The WebLogic managed server that was created for the RMS Webhelp application.
Example	rms_help_instance

Screen: Summary



All of the fields on this summary screen are read-only. In GUI mode of the installer, this screen provides the opportunity to review inputs and go back to previous screens to correct them if necessary.

Once you advance forward from this screen, the installer connects to the database and validates that the RMS user exists before beginning installation.

J

Appendix: Installer Silent Mode

Repeating an Installation Attempt

In addition to the GUI and text interfaces of the RMS installer, there is a silent mode that can be run. This mode is useful if you wish to run a repeat installation without retyping the settings you provided in the previous installation. It is also useful if you encounter errors in the middle of an installation and wish to continue.

The installer runs in two distinct phases. The first phase involves gathering settings from the user. At the end of the first phase, a properties file named ant.install.properties is created with the settings that were provided. Then the second phase begins, where this properties file is used to provide your settings for the installation.

To skip the first phase and re-use the ant.install.properties file from a previous run, follow these instructions:

1. Edit the ant.install.properties file and correct any invalid settings that may have caused the installer to fail in its previous run.
2. The installer screens remove any password properties from the ant.install.properties after they run. You may need to add these in to your properties file.
3. Look for duplicate properties in the ant.install.properties file. Some properties are set on multiple pages to ensure default values when a page is only displayed under certain conditions. For example, if there are two instances of input.property.name, remove all but the last one.
4. Run the installer again with the silent argument.

Example: install.sh silent.

K

Appendix: URL Reference

JDBC URL for a Database

Used by the Java application and by the installer to connect to the database.

Thick Client Syntax: jdbc:oracle:oci:@<sid>

<sid>: system identifier for the database

Example: jdbc:oracle:oci:@mysid

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

<host>: hostname of the database server

<port>: database listener port

<sid>: system identifier for the database

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

LDAP Derver 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

Example: ldap://myhost: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.

OracleAS:

Syntax: opmn:ormi://<host>:<port>:<instance>/<app>

<host>: hostname of the OracleAS environment

<port>: OPMN request port of the OracleAS environment. This can be found in the <ORACLE_HOME>/opmn/conf/opmn.xml file.

<instance>: Name of the OC4J instance running the application

<app>: Deployment name for the application.

Example: opmn:ormi://myhost:6003:rsm-oc4j-instance/rsm13

Note: The JNDI provider URL can have a different format depending on your cluster topology. Consult the Oracle Application Server documentation for further details.

WebSphere:
Syntax: iiop://<host>:<port>
<host>: hostname of the WebSphere environment
<port>: BOOTSTRAP port of the WebSphere server that is running the application.
Example: iiop://myhost:2809

L

Appendix: Common Installation Errors

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

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 to a Java runtime environment of version 1.4.2 or later and run the installer again.

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 Country and Currency Drop-Downs

Symptom

In GUI mode, when you click on the drop-down list selection for the primary country or currency, the list does not appear, and this message appears in the console window:

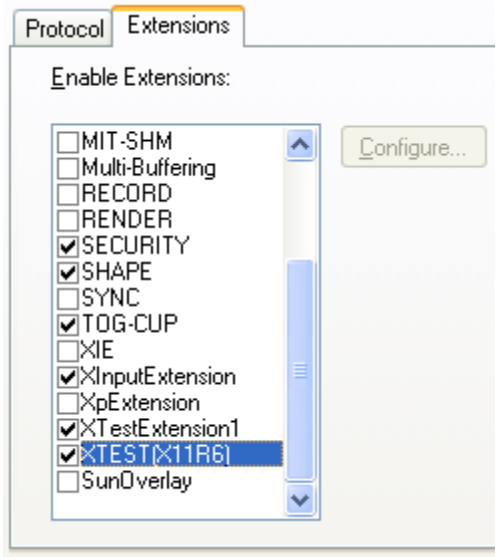
```
XTEST extension not installed on this X server: Error 0
```

Solution

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

To Enabling XTEST in Exceed, do the following.

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



5. Restart the X Server and re-run the RMS installer.

Could Not Execl Robot Child Process: Permission Denied

Symptom

When opening a drop-down list in GUI mode of the RMS 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.

For 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 as follows.

1. Bring down the RIB OAS in question.
2. Run /RIB_INSTALL_DIR>/InstallAndCompileAllRibOracleObjects.sql.
3. Run another object validate script (ex: inv_obj_comp.sql) to make sure objects are valid. (Some may have deal locked in the end of the previous step.)
4. Bring up the RIB OAS in question.

Error Connecting to Database URL

Symptom

After entering database credentials in the installer screens and hitting next, a message pops up with an error like this:

```
Error connecting to database URL <url> as user <user>
details...
```

The message prevents you from moving on to the next screen to continue the installation.

Solution

This error occurs when the installer fails to validate the user credentials you have entered on the screen. Make sure that you have entered the credentials properly. If you receive a message similar to this:

```
Error connecting to database URL <url> as user <user>
java.lang.Exception: UnsatisfiedLinkError encountered when using the Oracle
driver.
Please check that the library path is set up properly or switch to the JDBC thin
client.
```

It may mean that the installer is using the incorrect library path variables for the platform you are installing on. Open the file <STAGING_DIR>/rms/dbschema/common/preinstall.sh and make sure the variable use32bit is set to True if you are on a 32 bit platform, and False if you are on a 64 bit platform.

Multi-Threaded OCI Client Dumps Core after Reconnecting To Database

Symptom

If a multi-threaded Oracle client process that uses OCI to connect to a remote database loses connectivity with the database, it tries to reconnect and the client program continues to run. The program then dumps the core with the following stack trace, when Automatic Diagnostic Repository (ADR) is enabled.

```
skgfqio sdbgrfbibf_io_block_file dbgrfrbf_read_block_file dbgrmflrp_read_page
dbgrmblgmp_get_many_pages dbgrmmdrrmd_read_relation_meta_data
dbgrmmdora_open_record_access_full
dbgriporc_openrel_wcreate dbgrip_open_relation_access dbgrip_start_iterator
dbgrip_relation_iterator dbgruprac_read_adrctl...
```

Solution

Oracle Retail recommended you disable ADR (diag_adr_enabled=OFF, a sqlnet.ora parameter) while using multi-threaded OCI/OCCI application. diag_adr_enabled was introduced in Oracle 11g as a new method of tracing ADR. This will dump additional trace details.

Disabling 'diag_adr_enabled' does not disturb any functionality. Therefore, it can safely be unset by doing diag_adr_enabled=off in sqlnet.ora. However, if you still want tracing, you can have following parameters/variables set in sqlnet.ora:

```
trace_level_server=16 -- for server side NET tracing
trace_level_client=16 -- for client side NET tracing
```

For how to set traditional tracing, see the My Oracle Support document, “SQL*Net, Net8, Oracle Net Services - Tracing and Logging at a Glance” (ID 219968.1).

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 an error message on the screen saying ""no ocijdbc11 in java.library.path" that 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**. The installer will attempt to validate once again when installation starts, and the installer will fail if the credentials are incorrect.

Manually Running partition.ksh Script Fails

Symptom

The partition.ksh script is looking for rms13.tab, which is defined in partition_attributes.cfg. The files listed in the packaged location being referenced are named differently.

Solution

Modify two file values in STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/partition_attributes.cfg.

```
PREPARTDDL=./rms13.tab
PARTDDL=./rms13_part.tab
```

Should be valued

```
PREPARTDDL=./rms.tab
PARTDDL=./rms_part.tab
```

to feed the file names to the RMS installer properly.

Execute STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/part/partition.ksh at the UNIX command prompt. This script reads configuration information from the partition_attributes.cfg file and generates the partitioned DDL file STAGING_DIR/rms/dbschema/dbscripts_rms/ddl/rms_part.tab. This file is used later during the installation process.

M

Appendix: Manual Forms Compilation

The RMS Application installer must be used to configure an initial RMS installation. However after the initial install it may be necessary to compile forms (for example with a custom module or patch). Below are the instructions to manually compile the forms components of RMS.

It is assumed that Oracle WebLogic Server 11g Release 1 (10.3.3) has already been installed. If not, refer to "*Check Application Server Requirements*" in Chapter 1, "Preinstallation Tasks" before proceeding. Additionally, STAGING_DIR in this section refers to the directory created in "Create Staging Directory for RMS Application Files" in Chapter 4.

In order to use Forms Builder 11g for manual compilation of RMS 13 forms modules, Oracle Forms Services 11g Release 1 (11.1.1.3) must be used. Please refer to the Oracle Forms Services 11g Release 1 (11.1.1.3) documentation for the steps to manually compile objects.

> **Note:** It is necessary to have $ORACLE_HOME/network/admin/tnsnames.ora file configured in this WebLogic installation. Forms/reports will use this information for connectivity. Refer to Appendix B for an example setup of the tnsnames.ora file.

Set Environment Variables

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

1. The T2kMotif.rgb file that is sent out with WebLogic (10.3.3) must be modified. It located at the following location:
2. $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin
3. Make a copy of the file Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).
   Modify the file Tk2Motif.rgb file so that it contains the following line:
   ```
   Tk2Motif*fontMapCs: iso8859-2=UTF8
   ```
4. The Logon to the application server as a user with read and write access to the Weblogic files.
5. Set the DISPLAY variable to the IP address plus ":0.0" (ie: 10.1.1.1:0.0) of the application server.

6. Set the following variables:

   Notes:

   ORACLE_HOME is the location where WebLogic has been installed

   ORACLE_INSTANCE is the location where WebLogic has been installed and contains the executables to compile forms

   - All OS Platforms
     - 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=/@<Wallet Alias>
     - TK_ADMIN=/location/of /wallet/files
7. Copy $ORACLE_INSTANCE/config/FRComponent/frcommon/guicommon/tk/admin/Tk2Motif.rgb to $ORACLE_HOME/guicommon/tk/admin/Tk2Motif.rg

   Note: Please refer the document oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

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

   Example: /u00/oracle> sqlplus $UP

Compile RMS Toolset

To compile the RMS toolset, complete the following steps.

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

3. Run toolset.pll.sh to compile all Toolset .pll's.

> **Note:** If the toolset.pll.sh script is not used and the libraries are compiled individually, then they must be compiled in the following order (which is noted in the script):

- messge45.pll
- ariiflib.pll
- stand45.pll
- calend45.pll
- find45.pll
- item45.pll
- tools45.pll
- mblock45.pll
- mview45.pll
- nav45.pll
- work45.pll
- itnumtype.pll
- hierfilter.pll
- rmslib.pll
- cflex.pll

4. Check that each .pll file generated a .plx (to ensure that all .pll's compiled successfully).
5. Copy all forms (*.fmb files) in the INSTALL_DIR/base/toolset/src directory to the INSTALL_DIR/base/toolset/bin directory.
6. Change directories to INSTALL_DIR/base/toolset/bin.
7. Run forms.fm_fmb.sh to compile the Toolset reference forms.
8. Ensure all newly created fm_*.fmx files have been removed (reference forms should not have executable files).
9. Change directories to INSTALL_DIR/base/toolset/bin.
10. Run forms.fmb.sh to generate Toolset runtime forms – .fmx's.
11. Check that each non-reference form (.fmb file) generated a corresponding .fmx file.

> **Note:** Disregard fm_*.fmx files should they be created. These files should be removed. They should NOT exist in the INSTALL_DIR/base/toolset/bin directory.

12. Copy all menus (*.mmb files) in the INSTALL_DIR/base/toolset/src directory to the INSTALL_DIR/base/toolset/bin directory.
13. Change directories to INSTALL_DIR/base/toolset/bin.
14. Run menus.mmb.sh to generate Toolset runtime menus – .mmx's.
15. Check that each .mmb file generated a corresponding .mmx file.

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

Compile RMS Forms

To compile RMS forms, complete the following steps.

1. Copy all libraries (.pll files) in the INSTALL_DIR/base/rms/forms/src directory to the directories to the INSTALL_DIR/base/rms/forms/bin directory.
2. Change directories to INSTALL_DIR/base/rms/forms/bin.
3. Run forms.pll.sh to compile all RMS .pll's.
4. Check to make sure that each .pll file generated a corresponding .plx (to ensure that all .pll's compiled successfully).
5. Copy all forms (*.fmb files) in the INSTALL_DIR/base/rms/forms/src directory to the INSTALL_DIR/base/rms/forms/bin directory.
6. Change directories to INSTALL_DIR/base/rms/forms/bin.
7. Run forms.fm_fmb.sh to compile the RMS reference forms.
8. Ensure e all newly created fm_*.fmx files have been removed (reference forms should not have executable files).
9. Change directories to INSTALL_DIR/base/rms/forms/bin.
10. Run forms.fmb.sh to generate RMS runtime forms – .fmx's.
11. Check that each non-reference form .fmb file generated a corresponding .fmx file.

> **Note:** Disregard fm_*.fmx files should they be created. These files should be removed. They should NOT exist in the INSTALL_DIR/base/rms/forms/bin directory.

12. Copy all menus (*.mmb files) in the INSTALL_DIR/base/rms/forms/src directory to the INSTALL_DIR/base/rms/forms/bin directory.
13. Change directories to INSTALL_DIR/base/rms/forms/bin.
14. Run menus.mmb.sh to generate RMS runtime menus – .mmx's.
15. Check that each .mmb file generated a corresponding .mmx file.

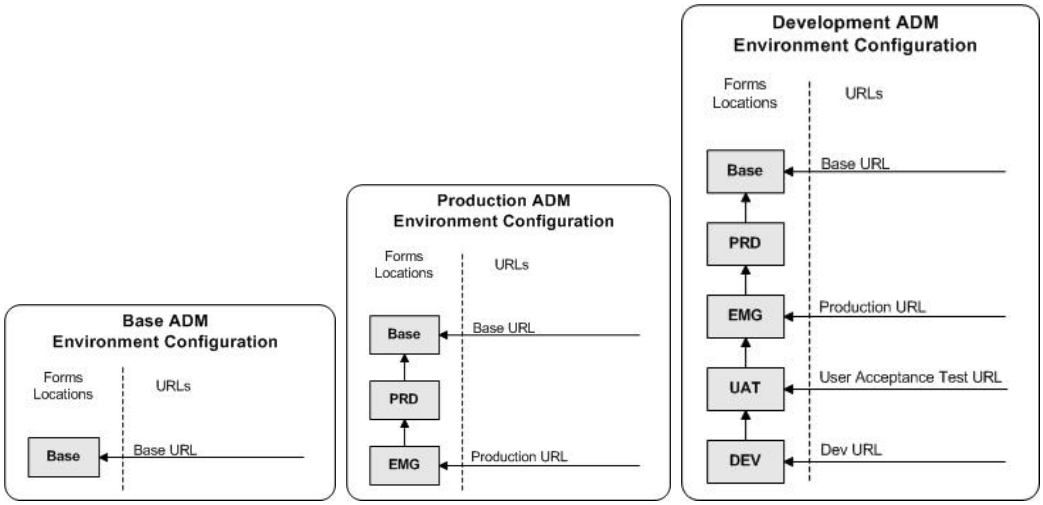
> **Note:** .err files may be created by the compilation scripts above. These files are logs of the compilation process and can be removed.

Appendix: Application Deployment Method

The RMS installer provides the option to configure multiple application deployment methods. This is a setup where there is still a single primary RMS installation, but there are additional levels where customization can occur. This means multiple URLs configured in formsweb.cfg with cascading FORMS_PATH values.

The installer provides three choices for cascading environment configuration:

- Base: A standard RMS base installation with one application installation folder, and one URL.
- Production: Base plus two additional forms directories for PRD and EMG and an additional URL for EMG.
- Development: Production plus two additional forms directories for UAT and DEV and two additional URLs for UAT and DEV.



The above diagrams show how the application deployment method environment configurations are set up in the forms installation.

The installer creates the set of URLs, and empty directories for the other environments. All forms installed by this installer are placed in the "Base" environment. We are simply laying down the structure for customizations and fixes that the user can make after installation is complete.

O

Appendix: Manual Batch Compilation

The RMS Application batch installer should be used to configure an initial RMS installation. However after the initial install it may be necessary to compile batch (for example with a custom module or patch). Below are the instructions to manually compile the batch components of RMS.

> **Note**: Warning messages may appear during the compilation of the batch. These warnings can be ignored if the batch executables are successfully generated.

Set Environment Variables

> **Note:** INSTALL_DIR is the location where RMS 13 batch was installed.

Make sure the following variables are set. The RMS batch installer should have created a batch.profile file located at INSTALL_DIR/batch.profile. This profile script can be used to set all of the environment variables listed below.

> **Example**: cd <INSTALL_DIR>
>
> . ./batch.profile

Variables set by batch.profile:

- PATH must include make, makedepend and the C compiler
- MMHOME=INSTALL_DIR/rms
- MMUSER=RMS Schema Owner
- ORACLE_HOME=Location of Oracle install
- ORACLE_SID=The Oracle Sid for the RMS database
- UP=/@<Wallet Alias>
- TNS_ADMIN=/location/of /wallet/files

> **Note:** Please refer the document oracle_wallet_setup_for_mom.doc" for instructions to setup database wallet.

> **Note:** Verify that TNS is set up correctly by using the UP variable to successfully log into the RMS 13 schema.
>
> **Example:** /u00/oracle> sqlplus $UP

AIX:

- LIBPATH=$ORACLE_HOME/lib:$MMHOME/oracle/lib/bin:$LDPATH
- OBJECT_MODE=64
- LINK_CNTRL=L_PTHREADS_D7

HP:

- SHLIB_PATH=$ORACLE_HOME/lib:$MMHOME/oracle/lib/bin:$SHLIB_PATH

Solaris:

- LD_LIBRARY_PATH=$ORACLE_HOME/lib:$MMHOME/oracle/lib/bin:$LD_LIBRARY_PATH

Linux:

- LD_LIBRARY_PATH=$ORACLE_HOME/lib:
$MMHOME/oracle/lib/bin:$LD_LIBRARY_PATH

Configure Make File

To configure the Make file, complete the following steps.

1. Log into the database server as a user that can connect to the RMS database
2. Change directories to INSTALL_DIR/oracle/lib/src
3. Several platform specific make files have been shipped with this release. Copy and rename the appropriate platform-specific make file to platform.mk

   Example: #cp platform_oel_64bit.mk platform.mk

4. Run the oramake script from INSTALL_DIR/oracle/lib/src directory. This uses the server's configurations to create a file called oracle.mk and copy an Oracle supplied make file (demo_rdbms.mk) to the lib/src directory.

Create Batch Libraries in Database

To create batch libraries in the database, complete the following steps.

1. Log into SQL*Plus as RMS13DEV and run the following commands. INSTALL_DIR is the location where the RMS batch files are installed (corresponds to $MMHOME). LIB_SUFFIX is the native library suffix for the operating system. (a for AIX, sl for HP, so for SUN and Linux).
   - CREATE OR REPLACE LIBRARY order_build_split_lib AS '<INSTALL_DIR>/oracle/lib/bin/libcreateord.<LIB_SUFFIX>';
   - CREATE OR REPLACE LIBRARY DEALINCLIB_C AS '<INSTALL_DIR>/oracle/lib/bin/libdealinc.<LIB_SUFFIX>';
   - CREATE OR REPLACE LIBRARY DEALORDLIB_C AS '<INSTALL_DIR>/oracle/lib/bin/libdealord. <LIB_SUFFIX>';
   - CREATE OR REPLACE LIBRARY scale_library AS '<INSTALL_DIR>/oracle/lib/bin/libsupcstrr. <LIB_SUFFIX>';
2. Exit SQL*Plus.

Revalidate RMS Database Objects

To revalidate RMS database objects, complete the following steps.

1. There is an invalid object compilation script included with the RMS database schema installer package (rms13dbschema.zip). Change directories to STAGING_DIR/rms/dbschema/dbscripts_rms/utility.
2. Log into SQL*Plus as RMS13DEV and run the following command.

   This script may need to be run more than once.

   ```
   SQL> @inv_obj_comp.sql
   ```

Compile Batch Libraries

To compile batch libraries, complete the following steps.

1. As user that can connect to the database, change directories to INSTALL_DIR/oracle/lib/src
2. To make library dependencies run one of the following commands.
   - For Linux use:

     ```
     make –f retek.mk –r depend 2>&1 | tee libdpnd.log
     ```
   - For other platforms use:

     ```
     make –f retek.mk depend 2>&1 | tee libdpnd.log
     ```

   Check the libdpnd.log file for errors.
3. To make batch libraries:
   - For Linux use:

     ```
     make –f retek.mk –r retek rms resa 2>&1 | tee libretek.log
     ```
   - For other platforms use:

     ```
     make –f retek.mk retek rms resa 2>&1 | tee libretek.log
     ```

   Check the libretek.log file for errors
4. To install batch libraries

   ```
   make –f retek.mk install
   ```

   The batch libraries should now be in INSTALL_DIR/oracle/lib/bin

Compile Batch Source Code

To compile batch sourc code, complete the following steps.

1. As user that can connect to the database, change directories to INSTALL_DIR/oracle/proc/src
2. To make dependencies run one of the following commands:
   - For Linux use:

     ```
     make –f mts.mk –r rms-depend recs-depend rtm-depend resa-depend 2>&1 | tee srcdpnd.log
     ```
   - For other platforms use:

     ```
     make –f mts.mk rms-depend recs-depend rtm-depend resa-depend 2>&1 | tee srcdpnd.log
     ```

   Check the srcdpnd.log file for errors

3. To make batch programs, run the following commands in the order stated:
   - For Linux use:

     ```
     make –f rms.mk –r PRODUCT_PROCFLAGS=dynamic=ansi ditinsrt
     make –f mts.mk –r rms-ALL recs-ALL resa-ALL rtm-ALL 2>&1 | tee srcall.log
     ```

   - For other platforms use:

     ```
     make –f rms.mk PRODUCT_PROCFLAGS=dynamic=ansi ditinsrt
     make –f mts.mk rms-ALL recs-ALL resa-ALL rtm-ALL 2>&1 | tee srcall.log
     ```

   Check the srcall.log file for errors.
4. To install batch programs:

   ```
   make –f mts.mk rms-install recs-install resa-install rtm-install
   ```

The batch programs should now be in INSTALL_DIR/oracle/proc/bin

P

Appendix: Oracle Single Sign-On

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

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

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

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

What Do I Need for Oracle Single Sign-On?

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

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

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

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

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

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

Oracle Single Sign-on Terms and Definitions

Authentication

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

Dynamically Protected URLs

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

Identity Management Infrastructure

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

MOD_OSSO

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

MOD_WEBLOGIC

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

Oracle Internet Directory

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

Partner Application

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

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

Realm

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

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

Statically Protected URLs

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

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

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

What Single Sign-On is not

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

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

How Oracle Single Sign-On Works

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

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

Statically Protected URLs

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

1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS).
2. The Oracle HTTP Server processes 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.

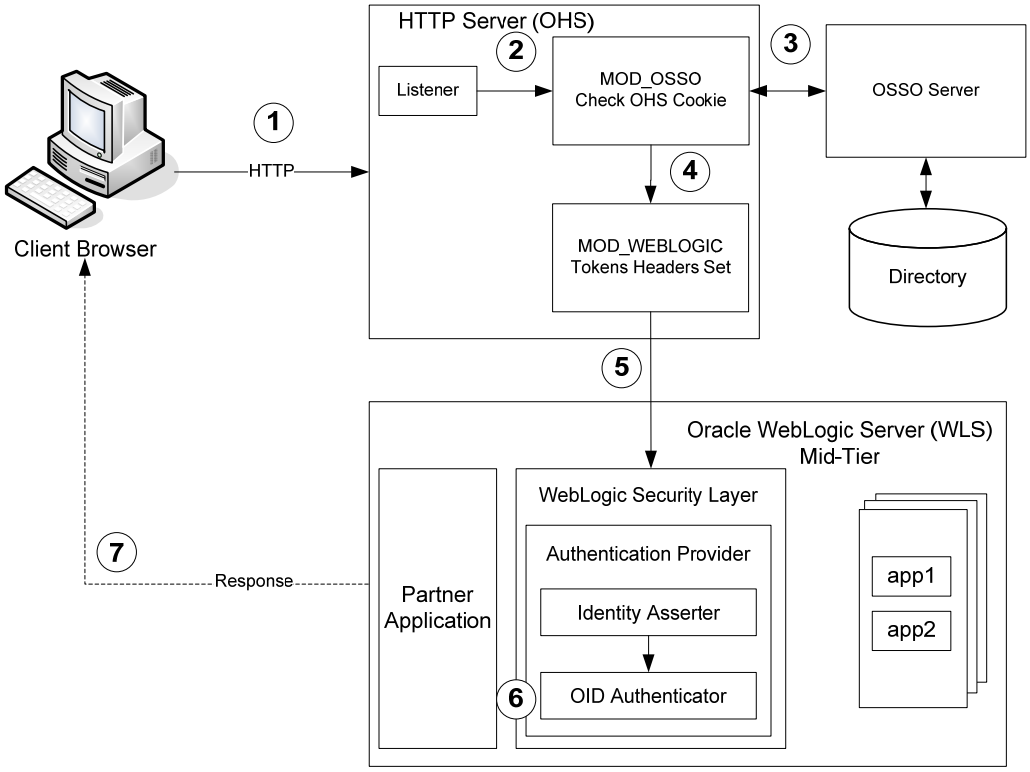
Dynamically Protected URLs

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 consists of installing the following components:

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

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

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

Infrastructure Installation and Configuration

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

OID User Data

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

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

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

OID with Multiple Realms

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

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

User Management

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

OID DAS

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

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

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

LDIF Scripts

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

User Data Synchronization

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

Q

Appendix: Single Sign-On Resource Access Descriptors

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

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

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

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

RADs may be created and via LDIF scripts as well. See My Oracle Support document 244526.1.

Appendix: AIX Shared Library Bug Fix

The env_rdbms.mk file for Oracle 11g has Bug #2143531. This bug was not fixed because there is a workaround. For the workaround, the following changes in **bold/italic** need to be made to the $ORACLE_HOME/rdbms/lib/env_rdbms.mk file. Notice that changes are made in both the BUILD_WITH_CONTEXT and BUILD_WITH_NO_CONTEXT functions.

```
-----------------------------------------
BUILDLIB_WITH_CONTEXT=generate_export_list() \
{ \
/bin/nm -X32_64 -B -h -g "$$1" | grep -v ' U ' | awk '{print $$3}' | \
egrep -v '^\.|^TOC' | sort | uniq ; \
}; \
generate_import_list() { \
LIB_NAME=$$1; \
IMP_FILE=$$2; \
\
cat ${ORACLE_HOME}/rdbms/lib/xa.imp | head -1 | awk '{print $$0, "." }' >
$${IMP_FILE}; \
/bin/nm -X32_64 -C -B -h -g $${LIB_NAME} | grep ' U ' | grep -v "::" | grep -v "("
| grep -v "\.cc" | awk '{print $$3}' | sed -e "s/\.//g
" | grep -v "^_" >> $${IMP_FILE}; \
}; \
\
generate_import_list ***"$(OBJS)"*** $(SHARED_LIBNAME).imp; \
generate_export_list $(OBJS) > $(SHARED_LIBNAME).exp; \
$(LD) -bnoentry -bM:SRE -bE:$(SHARED_LIBNAME).exp -bI:$(SHARED_LIBNAME).imp \
-o $(SHARED_LIBNAME) $(OBJS) -L$(ORACLE_HOME)/lib -lc_r -lm ***$(LLIBCLNTSH)***
***$(MATHLIB)***
-----------------------------------------
BUILDLIB_NO_CONTEXT=generate_export_list() \
{ \
/bin/nm -X32_64 -B -h -g "$$1" | grep -v ' U ' | awk '{print $$3}' | \
egrep -v '^\.|^TOC' | sort | uniq ; \
}; \
generate_import_list() { \
LIB_NAME=$$1; \
IMP_FILE=$$2; \
\
cat ${ORACLE_HOME}/rdbms/lib/xa.imp | head -1 | awk '{print $$0, "." }' >
$${IMP_FILE}; \
/bin/nm -X32_64 -C -B -h -g $${LIB_NAME} | grep ' U ' | grep -v "::" | grep -v "("
| grep -v "\.cc" | awk '{print $$3}' | sed -e "s/\.//g
" | grep -v "^_" >> $${IMP_FILE}; \
}; \
\
generate_import_list ***"$(OBJS)"*** $(SHARED_LIBNAME).imp; \
generate_export_list $(OBJS) > $(SHARED_LIBNAME).exp; \
$(LD) -bnoentry -bM:SRE -bE:$(SHARED_LIBNAME).exp -bI:$(SHARED_LIBNAME).imp \
-o $(SHARED_LIBNAME) $(OBJS) -L$(ORACLE_HOME)/lib -lc_r -lm ***$(LLIBCLNTSH)***
***$(MATHLIB)***
```

Appendix: Inserting New Languages

A new language was added for the RMS 13.2 release, Croatian (hr). As the RMS dbschema installer does not support inserting new languages that have not already been installed, this section documents how to manually insert new languages as either primary or secondary languages. These steps should be done after installing the RMS 13.2 upgrade.

In this section <lang> represents the two or three-letter code for the language you wish to insert. This is the list of supported codes and the languages they represent:

- de - German
- es – Spanish
- el - Greek
- fr – French
- hu – Hungarian
- hr – Croatian
- it – Italian
- ja – Japanese
- ko – Korean
- nl – Dutch
- pl - Polish
- ptb – Brazilian Portuguese
- ru – Russian
- sv – Swedish
- tr - Turkish
- zhs – Simplified Chinese
- zht – Traditional Chinese

Insert Secondary Language Data

To insert secondary language data, complete the following steps.

> **Note:** These scripts are only for customers who wish to have a primary language of English and a secondary language of any combination of the supported languages. The scripts are UTF-8 encoded. We recommend installing them into a database that has been set to UTF-8.

1. Change directories to STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/rms/lang/.
2. Set the sqlplus session so that the encoding component of the NLS_LANG is UTF8. For example AMERICAN_AMERICA.UTF8.

3. Log into sqlplus with the RMS schema and run the following command:

```
SQL> @rms1320_secondary_<lang>.sql
```

4. Check the log file rms1320_secondary_<lang>. log for any errors.

> **Note:** Multiple secondary languages can be added to a primary language install of English.

Insert Primary Language Data

To insert primary language data, complete the following steps.

> **Note:** These scripts are only for customers who wish to have a primary language of one of the non-English supported languages. Once you run one of these primary scripts, you will not be able to revert back to English as your primary language. The scripts are UTF-8 encoded. We recommend installing them into a database that has been set to UTF-8.

1. Change directories to STAGING_DIR/rms/dbschema/dbscripts_rms/required_patches/mom-dbpatch/13.2/rms/lang/.
2. Set the sqlplus session so that the encoding component of the NLS_LANG is UTF8. For example AMERICAN_AMERICA.UTF8.
3. Log into sqlplus with the RMS schema and run the following command:

```
SQL> @rms1320_primary_<lang>.sql
```

4. Check the log file rms1320_primary_<lang>. log for any errors.

> **Note:** Only one language can be set as the primary language for the system.

T

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`, then the argument to a program would be the following:

```
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:

Create a wallet using the following command:

```
kstore -wrl <wallet_location> -create
```

After you run the command, 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.*

1. 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.

2. Repeat step 2 for all the database user accounts.
3. 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
   ```

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

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

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

Setting Up Wallets for Database User Accounts

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

- For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI
- For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL

For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI

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.

Java wallets do not have a password to update their entries. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application, after you unzip <app>13application.zip, cd into <app>/application/retail-public-security-api/bin (for example,
mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/reim13/wallet/bin

or

Unzip to

reim/application/retail-public-security-api-bin to run the commands below to adminster java wallets. The application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.

There are two scripts relating to this at that folder: dump_credentials.sh and save_credentials.sh.

Dump_credentials.sh

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

```
Dump_credentials.sh <wallet location>
```

Example:

```
dump_credentials.sh
location:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_do
main/sim-client/csm

Retail Public Security API Utility
==========================================
Below are the credentials found in the wallet at the
location:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_do
main/retail/reim13/config
==========================================

Apapplication 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_credentials.sh

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

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

Example

```
mspdv351:[1033_WLS]
/u00/webadmin/mock132_testing/rtil/rtil/application/retail-public-security-
api/bin> save_credential.sh -l wallet_test -a myalias -p mypartition -u myuser

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

Enter password:
Verify password:
```

> **Note**: -p in the above command is for partition name. You must specify the proper partition name which is used in application code for each Java application.
>
> save_credentials.sh and dump_credentials.sh scripts are the same for all applications.

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 Tie Back 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/10.3.3/WLS/user_projects/domains/132_mck_soa
_domain/retail/reim13/config
csm.wallet.partition.name=reim
```

How does the Wallet Tie Back to Java pgm batch Use (such as REIM batch)?

Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser 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 for 13.2

RETL 13.2 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 loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (MMHOME)>/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile, execution	Installer	n/a	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile	Installer	n/a	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	n/a	<Db_Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rwms schema owner>	Compile forms, execute batch	Installer	n/a	Alias hard-coded by installer
RPM app	DB	<RPM batch install dir>/.wallet	n/a	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	
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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip13	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip13	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip13	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm13	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rpm13	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			rpm13	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			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	
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/config							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms 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>/sim-client/csm							
			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	

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

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
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	

U

Appendix: Installation Order

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

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

Enterprise Installation Order

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

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

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

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

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

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

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

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