Oracle® Retail Merchandising System

Installation Guide
Release 13.1.0.2

September 2009

ORACLE

Oracle® Retail Merchandising System Installation Guide, Release 13.1.0.2

Primary Author: Wade Schwarz

Contributors: Nathan Young

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server – Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning, Oracle Retail Demand Forecasting, Oracle Retail Regular Price Optimization, Oracle Retail Size Profile Optimization, Oracle Retail Replenishment Optimization applications.

(ii) 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.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

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

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by SAP and imbedded in Oracle Retail Store Inventory Management.

(vi) 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.

(vii) 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.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration 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

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*

Customer Support

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

https://metalink.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 Upgrade Documentation

If you are installing the application for the first time, you install either a base release (for example, 13.1) or a later patch release . If you are installing a software version other than the base release, be sure to read the documentation for each patch release (since the base release) before you begin installation. Upgrade documentation can contain critical information related to the base release and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site (with the exception of the Data Model which is only available with the release packaged code):

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

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

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."

> **Note:** This is a note. It is used to call out information that is important, but not necessarily part of the procedure.

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

A hyperlink appears like this.

1

Preinstallation Tasks

Note: The RMS installer provides the option to configure multiple application deployment methods. See Appendix N: Application Deployment Method to help determine your deployment approach.

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.

Check Database Server Requirements

General Requirements for a database server running RMS include:

Supported on:	Versions Supported:
Database Server OS	OS certified with Oracle Database 11gR1 Enterprise Edition. Options are: ▪ Oracle Enteprise Linux 5 Update 2 (OEL 5.2) for Linux x86-64 ▪ AIX 6.1 TL1
Database Server	Oracle Database 11g Release 1 Enterprise Edition (minimum 11.1.0.7 patchset required) with the following patches and components: Patches: ▪ 7036284 (LOADJAVA RUN IN A DV ENVIRONMENT CANNOT LOAD CLASSES WITH A NAME LONGER THAN 128) ▪ 7378322 (ORA-00600: internal error code, arguments: [6704], [1], [532241], [532237]) ▪ 6800649 – (AIX only) when non-oracle user uses client utilities sqlldr/sqlplus/impdp/expdp, core dump is generated. Need to "relink all" after applying the patch RAC only ▪ 7697360 ORA-00600: internal error code, arguments: [k2vcbk_6], Database crashed during transaction recovery. 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) ANSI compliant C compiler (certified with OS and database version) Perl compiler 5.0 or later x-Windows interface

Verify Single Sign-On

If a Single Sign-On is to be used, verify the Oracle Infrastructure Server 10g version 10.1.2.3 server has been installed. Verify the Mid-Tier server hosting Oracle Forms is registered with the Infrastructure Oracle Internet Directory.

Check Application Server Requirements

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

Supported on:	Versions Supported:
Application Server OS	OS certified with Oracle Application Server 10g version 10.1.2.3. & Oracle Business Intelligence Publisher 10.1.3.4 Options are: ▪ Oracle Enteprise Linux 5 Update 2 (OEL 5.2) for Linux x86-64 ▪ AIX 6.1 TL1
Application Server	Oracle Business Intelligence Publisher 10.1.3.4 Oracle Application Server Forms and Reports 10g version 10.1.2.3 Patches: ▪ 7379122 MLR ON TOP OF 10.1.2.3 FOR CPUOCT2008

Check Web Browser and Client Requirements

General requirements for client running RMS include:

Requirement	Version
Operating system	Windows 2000 or XP
Display resolution	1024x768
Processor	Pentium processor (minimum 450 MHz)
Memory	minimum of 256 MB RAM
Sun JRE Plug-in	1.4.1+
Microsoft Internet Explorer	version 6.0 and higher

Supported Oracle Retail Products

Product	Version
Oracle Retail Active Retail Intelligence (ARI)	13.1
Oracle Retail Price Management (RPM)	13.1
Oracle Retail Allocation	13.1
Oracle Retail Invoice Matching (ReIM)	13.1
Oracle Retail Store Inventory Management (SIM)	13.1
Oracle Retail Warehouse Management System (RWMS)	13.1
Oracle Retail Data Warehouse (RDW)	13.1
Oracle Retail Predictive Application Server (RPAS)	13.0.3

Supported Oracle Retail Integration Technologies

Integration Technology	Version
Oracle Retail Extract, Transform and Load (RETL)	13.1
Oracle Retail Integration Bus (RIB)	13.1
Oracle Retail Service Layer (RSL)	13.1

Create a UNIX User Account to Install the Software

The following user should be created on both the application and database servers.

1. Create a UNIX group named “dev”.
2. Create UNIX user named “oretail” and assign it to the “dev” group. This user will install the RMS software.

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 OAS and Database installations
- Real Application Cluster Database and Oracle Application Server Clustering

The Oracle Retail products have been validated against a 11.1.0.7 RAC database. When using a RAC database, all JDBC connections should be configured to use OCI connections rather than THIN connections. It is suggested that when using OCI connections, the Oracle Retail products database be configured in the tnsnames.ora file used by the Oracle Application Server installations.

Clustering for Oracle Application Server 10.1.3 is managed as an Active-Active cluster accessed through a hardware Load Balancer. It is suggested that a VirtualHost be added to the OAS 10.1.3 reflecting the Virtual Server Name configured in the load balancer. It is also suggested that the OC4J select method be configured to prefer the use of local OC4J instances. The Oracle Retail products are currently not validated to be distributable at the application level in an OAS 10.1.3 cluster.

Clustering for Oracle Application Server 10.1.2 is managed as an Active-Active cluster accessed through a hardware Load Balancer. It is suggested that the Web Cache installation included with OAS 10.1.2 be configured to reflect all application server Mid-Tier installations. Validation has been completed utilizing a RAC 11.1.0.7 Oracle Internet Directory database with the OAS 10.1.2 cluster.

References for Configuration:

- Oracle® Application Server High Availability Guide 10g Release 3 (10.1.3) Part Number B15977-02
- Oracle® Application Server High Availability Guide 10g Release 2 (10.1.2) Part Number B14003-05
- Oracle Real Application Clusters Administration and Deployment Guide 11g Release 1 (11.1) Part Number B28254-07

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.

3

Database Installation Tasks—Full

RMS 13.1 Full Release

RMS 13.1 is a full baseline installation. The RMS 13.0 software should not be installed if installing from this release.

RMS Database Schema Distribution – Oracle Retail Applications Included

The RMS 13.1 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

1. Log into the database server as oretail.
2. Create a staging directory for the RMS database schema installation software. There should be a minimum of 180 MB disk space available in this location.
3. Copy the rms13dbschema.zip file from the RMS 13.1 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 it's entirety before proceeding with the installation.

Sample Partitioning

The RMS 13.1 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>, e.g., item_loc_hist.eow_date.date. When placing 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

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 1, with appropriate patches, has already been installed. If not, refer to "*Check Database 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 Database Files", Chapter 1.

Please review the "Establish 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 Appendixes A, B, C, D, and E.

Create the Database as Follows:

1. Login to UNIX as the oracle user; typically the user that owns the Oracle Database software.
2. Create the Oracle recommended OFA directory structure for the database (datafile directories, adump, bdump, cdump, arch, create, exp, pfile, udump, utl_file_dir)
3. Place an entry in the oratab file for the database and execute oraenv to set the ORACLE_SID and ORACLE_HOME environment variables.
4. Create a staging directory for the RMS database hotfix.
5. Copy the zip file for the RMS 13.1 hotfix to the staging directory. This is referred to as HOTFIX_DIR when installing database software.
6. Change directories to HOTFIX_DIR and extract the zip file.
7. Copy HOTFIX_DIR/create_db/init.ora to the $ORACLE_HOME/pfile directory and rename it to init${ORACLE_SID}.ora. Modify the parameters according to guidelines specified in this file.
8. Create a symbolic link from $ORACLE_HOME/pfile/init${ORACLE_SID}.ora to $ORACLE_HOME/dbs/init${ORACLE_SID}.ora.
9. Modify the HOTFIX_DIR/create_db/crdb1.sql file. Refer to comments in this file regarding modifications that need to be made.
10. Login to SQL*Plus as SYSDBA and execute HOTFIX_DIR/create_db/crdb1.sql. Review crdb1.log for errors and correct as needed.
11. Login to SQL*Plus as SYSDBA and execute HOTFIX_DIR/create_db/crdb2.sql. Review crdb2.log for errors and correct as needed.
12. Login to SQL*Plus as SYSDBA and execute HOTFIX_DIR/crdb3.sql. Review JServer.log, context.log and xdb_protocol.log for errors and correct as needed.
13. Configure the listener. The RMS application uses external procedure calls. Therefore, the listener.ora and tnsnames.ora files must be configured properly. Refer to Appendix B.

Create the Tablespaces:

Modify HOTFIX_DIR/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 HOTFIX_DIR/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 HOTFIX_DIR/create_db/create_rms_tablespaces.sql 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 D 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/dbscripts_rms/utility
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_user.log for any errors. This log file should be removed to prevent the password from being compromised.

Create the Database User for Allocation

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.sql script located here:

STAGING_DIR/rms/dbschema/dbscripts_rms/allocation/utility/create_user.sql

Example: ALLOC13DEV

Review RIB CLOB Settings

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.

For more information on the RIB objects see the RIB documentation.

Run the RMS Database Schema Installer

Note: Appendix H contains 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 M: 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]

   Depending on system resources, a typical installation takes anywhere from 30 minutes to two hours.

   The RMS 13.1 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 13.1 installation select the "Full" option on the "Full Install or Patch Option" screen. RMS 13.1.x patches released after RMS 13.1 will utilize the Patch option.
6. 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.

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

   ```
   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 also writes 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 and run the installer again. You can run the installer in silent mode so that you don't have to go through the installer screens again. See Appendix K of this document for instructions on silent mode.

See Appendix M of 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. If you have dropped the RMS schema and want to start with a clean install, you can delete the .dbhistory file so that the installer runs through all of the scripts again. 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 executing the standard SQL "create user" command. Once these users have been created, execute the following to grant proper privileges for these users.

   ```
   grant create session, create table, create procedure,create view, delete any table, insert any table, select any table, update any table, select any sequence, execute any procedure, create any procedure, drop any procedure, execute any procedure, create any table, create job, drop any table to <userid>;
   grant developer to <userid>;
   ```

   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 that the user has access to.
3. Run the following scripts as the new user to give new users security privileges.

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

   These scripts can be found in the RMS database schema installer package under rms/dbschema/dbscripts_rms/utility.

4

Batch Installation Tasks—Full

Create Staging Directory for RMS Batch Files

1. Log into the database server as oretail.
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 30 MB disk space available in this location.
3. Copy the rms13batch.zip file from the RMS 13. 1 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:** Appendix I 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 M: 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.

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

   ```
   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 $MMHOME/rfx.

See Appendix E of this document for more information about RETL.

Data Conversion Scripts

The RMS batch installer installs the data conversion scripts under $MMHOME/external/scripts. To complete the setup of these files, perform the following steps.

1. Create the following new directories:

   INSTALL_DIR/external/data

   INSTALL_DIR/external/logs

The RMS Batch installer should have already created INSTALL_DIR/scripts.

2. 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.

3. Log into sqlplus as SYSTEM and grant access to them by running the following commands:

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

5

Application Server Installation Tasks—Full

It is assumed that Oracle Application Server 10g version 10.1.2.3 (OAS) 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 Forms Builder 10g for manual compilation of RMS 13 forms modules, Oracle Developer Suite 10g Release 2 (10.1.2.3) must be used. Please refer to the Oracle Developer Suite 10g Release 2 documentation for the steps to manually compile objects.

> **Note:** It is necessary to have $ORACLE_HOME/network/admin/tnsnames.ora file configured in this OAS 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 Oracle Application Server 10g (10.1.2.3) has been installed

The T2kMotif.rgb file that is sent out with Oracle Application Server 10g (10.1.2.3) must be modified. It is located at the following location:

$ORACLE_HOME/guicommon/tk/admin

1. Make a copy of the file ORACLE_HOME/guicommon/tk/admin/Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).
2. Modify the file Tk2Motif.rgb file so that it contains the following line:

```
Tk2Motif*fontMapCs: iso8859-2=UTF8
```

Create a New OC4J Instance for Webhelp

The RMS Webhelp application must be deployed to its own dedicated OC4J instance. For instructions on how to create a new OC4J instance, see *Creating OC4J Instances on the Application Tier* in the *Installing and Configuring the myJ2EECompany Application Infrastructure* chapter of the *Oracle Application Server Enterprise Deployment Guide*.

1. Log into the server which is running your RMS installation. Set your ORACLE_HOME environment variable to point to this installation.
2. Choose a name for the new OC4J instance.

   Example: rms-help

3. Create this OC4J instance as documented in the *Oracle Application Server Enterprise Deployment Guide*.

4. Start the OC4J instance. You can do this through the Enterprise Manager web interface, or on the command line using the opmnctl utility:

   Example: $ORACLE_HOME/opmn/bin/opmnctl @cluster startproc ias-component=rms-help

5. Verify that the OC4J group was fully started. If you are using the Enterprise Manager web interface, the instance should have a green arrow indicating that it is running. On the command line, verify that the instance has a status of "Alive".

   Example: $ORACLE_HOME/opmn/bin/opmnctl status

Create Staging Directory for RMS Application Server Files

1. Log into the application server as the oretail user.
2. Create a staging directory for the RMS application installation software. There should be a minimum of 500 MB disk space available in this location.
3. Copy the file rms13appserver.zip from the RMS 13. 1 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 rms13appserver.zip. This will create an rms/application subdirectory under STAGING_DIR.

Run the RMS Application Installer

Note: Appendix **Error! Reference source not found.** contains details on every screen and field in the application installer.

1. Logon to your application server as the oretail user.
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
ORACLE_HOME	The location where Oracle Application Server 10g (10.1.2.3) has been installed.	ORACLE_HOME= /u00/webadmin/product/OAS/myversion/midtier export ORACLE_HOME
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
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 need to be using 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]

   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.
6. The RMS Application installer might launch the Retail OCM Installer automatically after it is finished with the RMS installation. The Retail OCM Installer can only be run if the UNIX account owning the installer process has write access to the $ORACLE_HOME directory. If you do not have such access, it is safe to opt out of the OCM install by clicking the Cancel button in the Retail OCM Installer. See the *Oracle Configuration Manager* section later in this chapter for more details on OCM.
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.

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

   ```
   chmod 600 ant.install.properties
   ```
8. 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:

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

/u00/webadmin/product/10.1.2.3_FULL/midtier/Apache/Apache/conf/httpd.conf
/u00/webadmin/product/10.1.2.3_FULL/midtier/forms/java/oracle/forms/registry/R
egistry.dat
/u00/webadmin/product/10.1.2.3_FULL/midtier/forms/server/formsweb.cfg
/u00/webadmin/product/10.1.2.3_FULL/midtier/forms/admin/resource/US/fmrweb.res
/u00/webadmin/product/10.1.2.3_FULL/midtier/forms/admin/resource/US/fmrweb.res
_utf8.res
Have the Oracle administrator copy everything in
/projects/rmsse/con/installs/app/post
to /u00/webadmin/product/10.1.2.3_FULL/midtier to update the files,
and then restart the application server for the changes to take effect.

example: cp -R * /u00/webadmin/product/10.1.2.3_FULL/midtier
```

Resolving Errors Encountered During Application Installation

In the event a form or menu does not compile, go to <INSTALL_LOCATION>/base/error and see which objects didn't compile. To try and manually recompile the object run <INSTALL_LOCATION>/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.

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 whose ORACLE_HOME you used for the RMS application installer is referred to as the *master node*. All other nodes are referred to as the *remote nodes*.

1. To complete the RMS forms application install, the installer provided new versions of formsweb.cfg and the newly-created env file(s) 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 node(s).

   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://metalink.oracle.com`

Oracle Configuration Manager Installer Guide (Doc ID: 835024.1)

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

OCM Documentation Link

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

RMS Reports Installed by the Application Installer

The application installer installs RMS report files to $MMHOME/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; Doc ID 838623.1 on My Oracle Support (formerly MetaLink).

6

RMS Reports—Full

Verify that Oracle BI Publisher has been set up correctly; refer to the *RMS Operations Guide Volume 3*.

1. Click on the Admin tab and then click Report Repository under System Maintenance. The Path variable should be set as part of the BI Publisher install, REPORTS_DIR.
2. In the default.env file located at ORACLE_HOME/forms/server/ add the following values (this file may be renamed rms13.env):
   - ORACLE_RMS_REPORTS_HOST=http://<server>:<port>:<context root for reports>
   - ORACLE_RMS_REPORTS_SERVER=<context root for reports>
   - ORACLE_RMS_RWSERVER=/<view userid>/
3. Go to the REPORTS_DIR and create a folder with the same name as ORACLE_RMS_RWSERVER.
4. Copy the files and directories from STAGING_DIR/rms/application/rms13/reports/* to folder created in step 3.

Part II: Upgrade Installation

The database portion of RMS can be upgraded from release 13.0.2 to release 13.1. Part II of this guide details the steps needed to perform an upgrade installation of RMS. For additioinal information on the upgrade, see the *Oracle Retail Upgrade Guide* (Doc ID 837368.1) 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.

7

RMS Database Installation—Upgrade

There are two different methods to use for installing the RMS 13.1 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 13.1 release. Any patches that were released prior to 13.1 (For example, 13.0.1 and 13.0.2) will not be compatible with this installer. If you need to upgrade from 13.0.1 to 13.1, please use the 13.0.1 installer to apply the 13.0.2 patch, and the 13.1 installer to apply the 13.1 patch.

Option 1: Upgrade RMS Database using the Installer

The RMS 13.1 database schema installer may be used to apply the RMS upgrade. The entire 13.1 RMS upgrade may be installed by re-running the installer used with the RMS 13.1 full release.

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 13.1 database schema installer was originally expanded. The installer files from the original RMS 13.1 installation can be re-used or a new directory can be created with a fresh copy of the RMS 13.1 database schema installer.

Before you apply the RMS 13.1 upgrade:

- Make a backup of all your objects and database schema.
- Check that RMS 13.0.2 is installed.
- Review the RMS 13.1 Release Notes (rms-131-rn.pdf).

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.

Grant Permissions to RMS Schema

The RMS schema will need certain privelages granted to it. Log into the database as SYSTEM with sqlplus and grant these privelages by running the following command:

```
SQL> grant create job to <RMS schema>;
```

Replace <RMS schema> with the name of your RMS schema.

Create Staging Directory for RMS Database Schema Files

1. Log into the database server as oretail.
2. Create a staging directory for the MOM 13.1 Upgrade. There should be a minimum of 50 MB disk space available in this location.
3. Copy the mom-dbpatch.zip file from the RMS 13.1 release to the staging directory. This is referred to as DB_PATCH_DIR when upgrading a database schema.
4. Change directories to DB_PATCH_DIR and extract the mom-dbpatch.zip file. This creates an mom-dbpatch/13.1 subdirectory under DB_PATCH_DIR
5. Create a staging directory for the RMS database schema installation software. There should be a minimum of 180 MB disk space available in this location.
6. Copy the rms13dbschema.zip file from the RMS 13.1 release to the staging directory. This is referred to as STAGING_DIR when upgrading a database schema.
7. 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

Note: Appendix H contains 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. 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 M: 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 13.1 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 DB_PATCH_DIR/mom-dbpatch/<product>. This directory should contain a <product>_controller.ksh file (for example, rms_controller.ksh), which the installer runs to apply the RMS 13.1 upgrade.

8. After the installer is complete, you can check its log file: rms-install-dbschema.<timestamp>.log.

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

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

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 13.1 upgrade:

- Make a backup of all your objects and database schema.
- Check that RMS 13.0.2 is installed.
- Review the RMS 13.1 Release Notes (rms-131-rn.pdf).

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.

Grant Permissions to RMS Schema

The RMS schema will need certain privelages granted to it. Log into the database as SYSTEM with sqlplus and grant these privelages by running the following command:

```
SQL> grant create job to <RMS schema>;
```

Replace <RMS schema> with the name of your RMS schema.

Create Staging Directory for RMS Database Schema Files

1. Log into the database server as oretail.
2. Create a staging directory for the MOM 13.1 Upgrade. There should be a minimum of 50 MB disk space available in this location.
3. Copy the mom-dbpatch.zip file from the RMS 13.1 release to the staging directory. This is referred to as DB_PATCH_DIR when upgrading a database schema.
4. Change directories to DB_PATCH_DIR and extract the mom-dbpatch.zip file. This creates an mom-dbpatch/13.1 subdirectory under DB_PATCH_DIR

Run the RMS Database Controller Scripts

1. Change directories to DB_PATCH_DIR/mom-dbpatch/13.1.
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 DB_PATCH_DIR/mom-dbpatch/13.1/<product>/templates/controller.cfg to DB_PATCH_DIR/mom-dbpatch/13.1/<product>/controller.cfg
   - Open the controller.cfg file you just created and replace the tokens for the following variables with the appropriate values:
     i. Export PATCH_DIR=DB_PATCH_DIR/mom-dbpatch/13.1/<product>
     ii. export SCHEMA_OWNER=<The name of the RMS schema>
     iii. 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
     iv. export PASSWORD=<password for the MMUSER schema>
     v. export ORACLE_SID=<SID for the database the MMUSER schema resides in>
6. The product upgrades should be run in the following order: RMS, RPM, ReIM, Alloc_RMS, and Allocation. The Alloc controller is used to apply the necessary Allocation 13.1 upgrades to the Allocation schema, while the Alloc_RMS controller is used to apply the necessary Allocation 13.1 upgrades to the RMS schema. While you can choose not to run any of the upgrades, all of the patches other than RMS depend on the RMS 13.1 Upgrade being run. If you upgrade RMS you should also upgrade

RPM; there is also a dependancy between Alloc_RMS/Alloc. For each product you wish to upgrade, cd to DB_PATCH_DIR/mom-dbpatch/13.1/<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.

7. If the installation fails for any of the upgrades before completion, look at the logs in the DB_PATCH_DIR/mom-dbpatch/13.1/<product>/error and DB_PATCH_DIR/mom-dbpatch/13.1/<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 DB_PATCH_DIR/mom-dbpatch/13.1/<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 DB_PATCH_DIR/mom-dbpatch/13.1/<product>/processed directory.

8

Batch Installation Tasks—Upgrade

There are two different methods to use for installing the RMS 13.1 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 13.1 Batch upgrade:

- Make a backup of all your Batch files.
- Review the RMS 13.1 Release Notes (rms-131-rn.pdf).

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 Chapter titled "Batch Installation Tasks – Full" 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

1. Log into the database server as oretail.
2. Create a staging directory for the RMS 13.1 Batch Upgrade. There should be a minimum of 10 MB disk space available in this location.
3. Copy the rms131batchpatch.zip file from the RMS 13.1 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 rms131batchpatch.zip file. This creates batch/lib and batch/proc subdirectories under BATCH_PATCH_DIR.

Set Environment Variables

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

Make sure the following variables are set. The RMS 13.1 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
- PASSWORD=RMS Schema Owner Password
- ORACLE_HOME=Location of Oracle install
- ORACLE_SID=The Oracle Sid for the RMS database

AIX:

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

Oracle Enterprise Linux:

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

Compile Batch Libraries

1. Copy the files from BATCH_PATCH_DIR/batch/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

1. Copy the files from BATCH_PATCH_DIR/batch/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.

9

Application Server Installation Tasks—Upgrade

There are two different methods to use for installing the RMS 13.1 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 13.1 upgrade:

- Make a backup of all your forms and library files.
- Review the RMS 13.1 Release Notes (rms-131-rn.pdf).

Run the RMS Application Installer

The steps to run the Application installer for an upgrade are the same as the steps for a full install. See the Chapter titled "Application Server Installation Tasks – Full" for details.

Modify OAS Files

1. Open the RMS *.env file under ORACLE_HOME/forms/server, and make sure the variable FORMS_USERNAME_CASESENSITIVE=1 exists and is set properly.
2. Open the file formsweb.cfg located at ORACLE_HOME/forms/server. In the RMS environment section, make sure the following variables are commented out: baseHTMLjinitiator, jpi_download_page, jpi_classid, jpi_codebase, jpi_mimetype.

Example:

```
## baseHTMLjinitiator=basejpi.htm
##
jpi_download_page=http://java.sun.com/products/archive
/j2se/1.4.2_06/index.html
## jpi_classid=clsid:CAFEEFAC-0014-0002-0006-
ABCDEFFEDCBA
##
jpi_codebase=http://java.sun.com/products/plugin/autodl
/jinstall-1_4_2-windows-i586.cab#Version=1,4,2,06
## jpi_mimetype=application/x-java-applet;jpi-
version=1.4.2_06
```

Option 2: Compile RMS Toolset and Forms Directly

Create Staging Directory for RMS Application Upgrade Files

1. Log into the application server as the oretail user.
2. Create a staging directory for the RMS application installation software. There should be a minimum of 250 MB disk space available in this location.
3. Copy the file rms131apppatch.zip from the RMS 13. 1 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 rms13appserver.zip. This will create an base, reports, and webhelp subdirectories under APP_PATCH_DIR.

Set Environment Variables

Note: INSTALL_DIR is the location where RMS 13 forms were installed.
ORACLE_HOME is the location where Oracle Application Server 10g (10.1.2.3) has been installed.

Make sure the following variables are set. The RMS 13.1 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**
  - 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_HOME/guicommon/tk/admin
  - UP=<RMS schema owner>/<RMS schema password>@<RMS database>

    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
- **Oracle Enterpirse Linux**
  - LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/jdk/jre/lib

RMS Toolset Installation

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/base/toolset into INSTALL _DIR/base/toolset and APP_PATCH_DIR/base/forms into INSTALL _DIR/base/forms.
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 pll2plx10gr2_toolset to compile all Toolset .pll's.

   Note: If the pll2plx10gr2_toolset script is not used and the libraries are compiled individually, then they must be compiled in the following order (which is noted in the pll2plx10gr2_toolset 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
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 fmb2fmx10gr2_fm (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 fmb2fmx10gr2 (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 mmb2mmx10gr2 (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.

RMS Forms Installation

1. Copy all the files from APP_PATCH_DIR/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 pll2plx10gr2_forms 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 fmb2fmx10gr2_fm (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 fmb2fmx10gr2 (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 mmb2mmx10gr2 (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.

Modify OAS Files

1. Open the RMS *.env file under ORACLE_HOME/forms/server, and make sure the variable FORMS_USERNAME_CASESENSITIVE=1 exists and is set properly.
2. Open the file formsweb.cfg located at ORACLE_HOME/forms/server. In the RMS environment section, make sure the following variables are commented out: baseHTMLjinitiator, jpi_download_page, jpi_classid, jpi_codebase, jpi_mimetype.

> **Example:**
> ## baseHTMLjinitiator=basejpi.htm
> ##
> jpi_download_page=http://java.sun.com/products/archive/j2se/1.4.2_06/index.html
> ## jpi_classid=clsid:CAFEEFAC-0014-0002-0006-ABCDEFFEDCBA
> ##
> jpi_codebase=http://java.sun.com/products/plugin/autodl/jinstall-1_4_2-windows-i586.cab#Version=1,4,2,06
> ## jpi_mimetype=application/x-java-applet;jpi-version=1.4.2_06

Create a New OC4J Instance for Webhelp

The RMS Webhelp application must be deployed to its own dedicated OC4J instance. For instructions on how to create a new OC4J instance, see *Creating OC4J Instances on the Application Tier* in the *Installing and Configuring the myJ2EECompany Application Infrastructure* chapter of the *Oracle Application Server Enterprise Deployment Guide*.

1. Log into the server which is running your RMS installation. Set your ORACLE_HOME environment variable to point to this installation.

2. Choose a name for the new OC4J instance.

   Example: rms-help

3. Create this OC4J instance as documented in the *Oracle Application Server Enterprise Deployment Guide*.
4. Start the OC4J instance. You can do this through the Enterprise Manager web interface, or on the command line using the opmnctl utility:

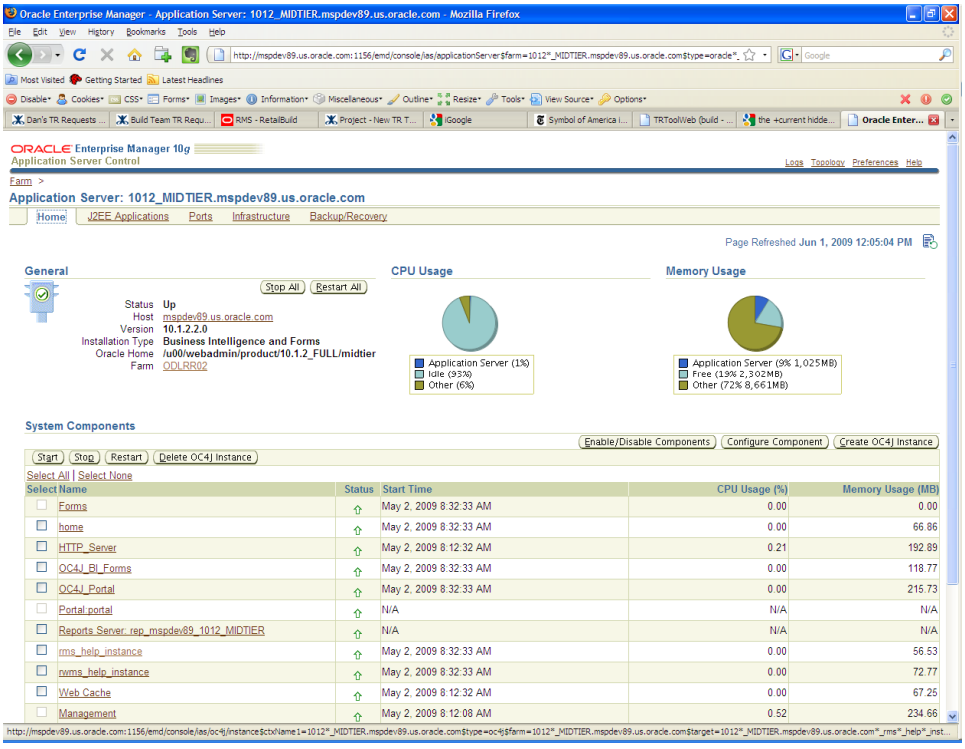
   Example: $ORACLE_HOME/opmn/bin/opmnctl @cluster startproc ias-component=rms-help

5. Verify that the OC4J group was fully started. If you are using the Enterprise Manager web interface, the instance should have a green arrow indicating that it is running. On the command line, verify that the instance has a status of "Alive".

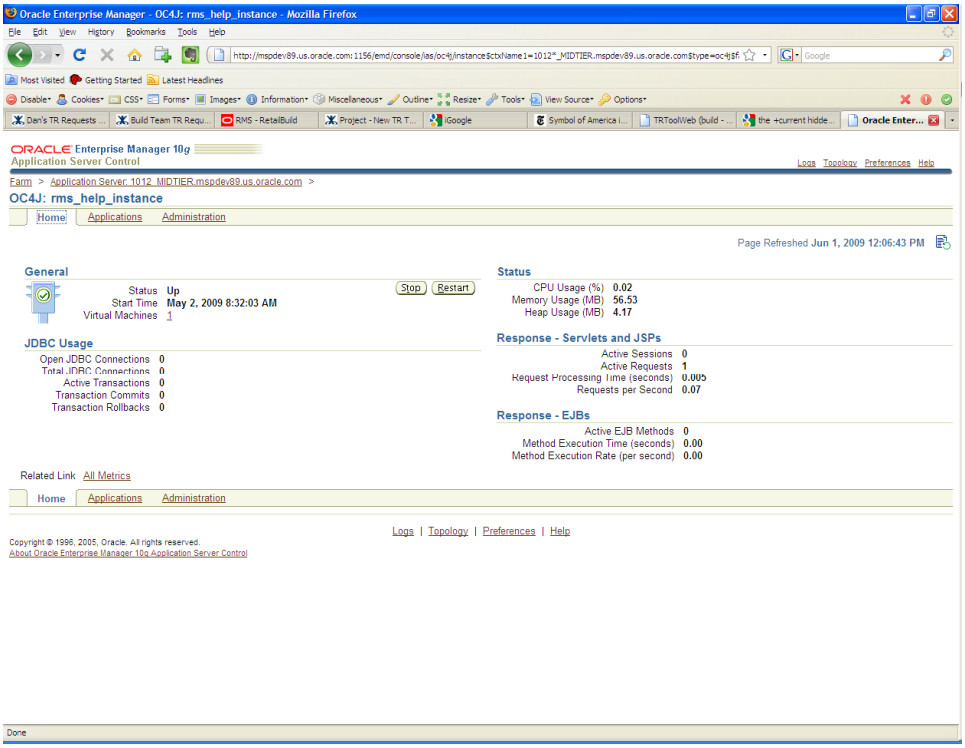
   Example: $ORACLE_HOME/opmn/bin/opmnctl status

Update Helpfile Installation

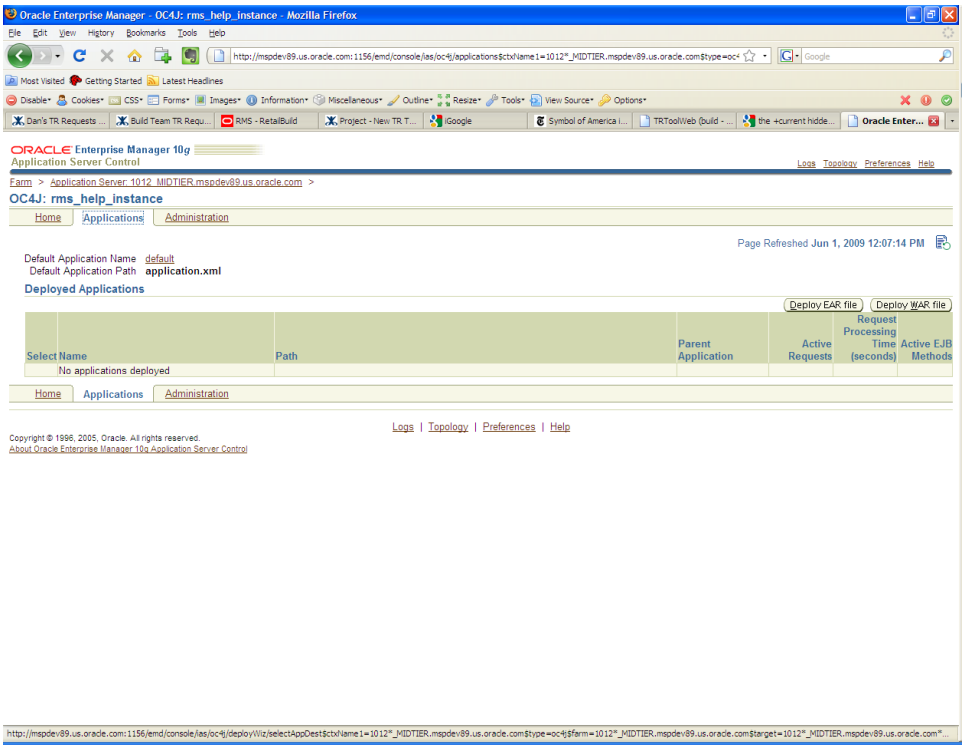
1. Log into the Enterprise Manager for the 10.1.2 OAS instance to which online help will be installed.



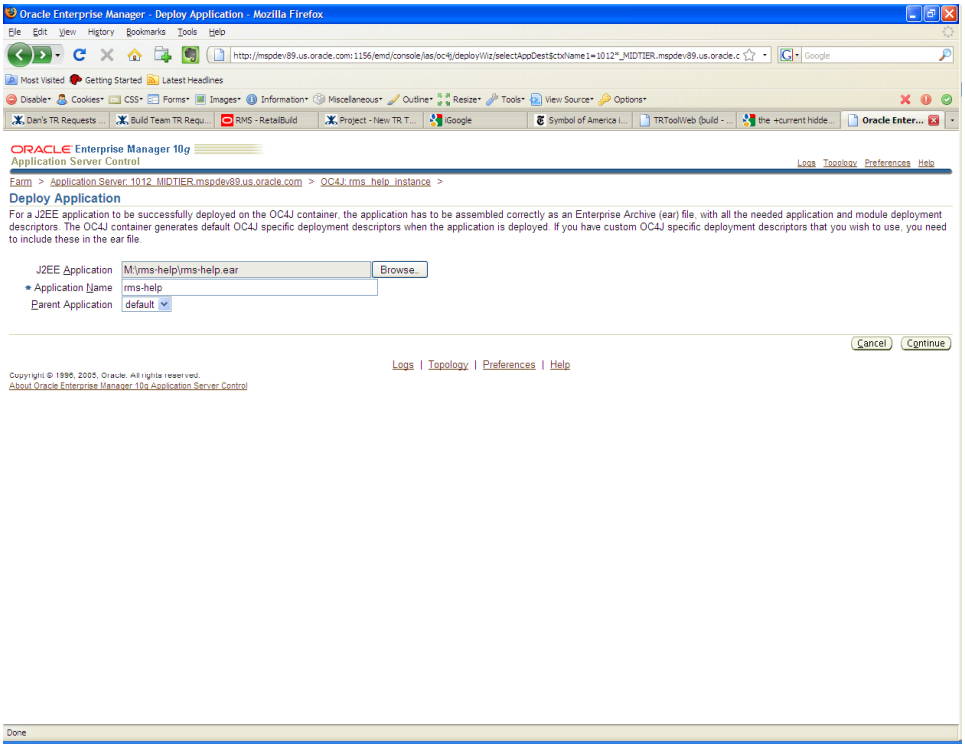
2. Click on the OC4J instance to which online help will be deployed. In this example **rms_help_instance** is being used.



3. Click on the **Applications** tab.

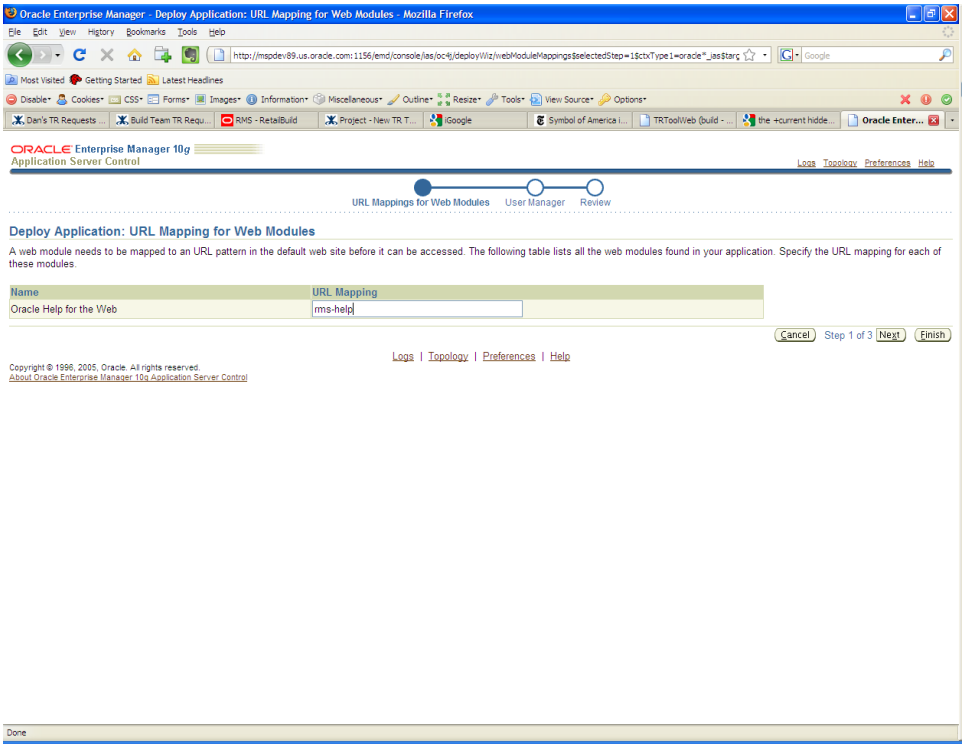


4. Click the **Deploy EAR File** button.
5. Click in the box entitled **J2EE Application** and browse to APP_PATCH_DIR/webhelp/rms-help.ear the ear file that will be deployed.
6. Enter **rms-help** for the Application Name, and click Continue.

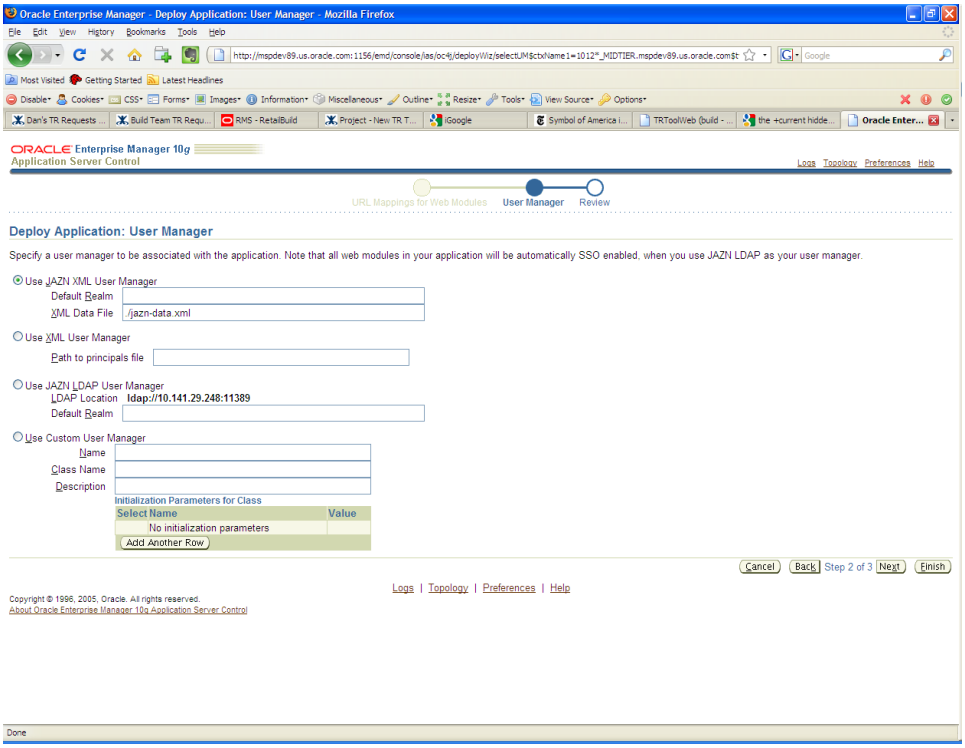


7. Enter **rms-help** in the URL-Mapping text box. This sets the context root for the online help files. Click Next.

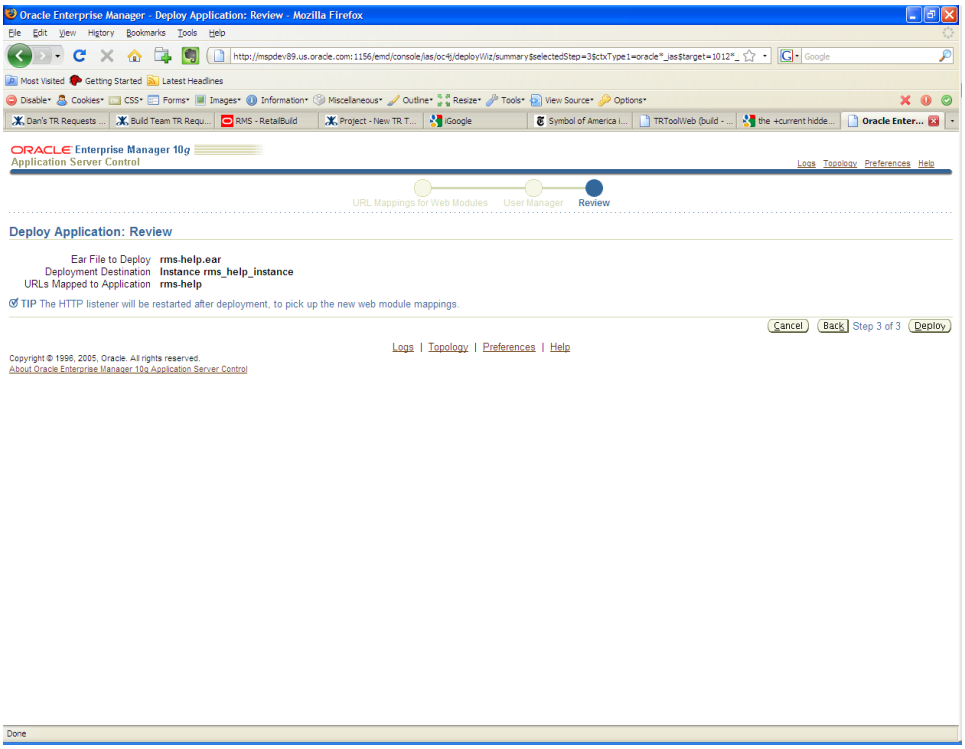
> **Note:** Using anything other than **rms-help** will prevent the rms application from accessing the help files



8. Leave the default values in place, and click Next.



9. Review the deployment properties, and click Deploy.



10. A progress screen appears while the application is being deployed.

11. When the ear file has been successfully deployed, click OK.

10

RMS Reports Installation—Upgrade

RMS Reports are included in the RMS Application upgrade: rms131apppatch.zip in the reports directory. To install the reports files, copy them from the RMS application upgrade APP_PATCH_DIR/reports to the reports directory created during RMS installation.

11

Data Migration

Included in the 13.1 release is a tool responsible for upgrading prexisting data in the RMS schema to the 13.1 schema once 13.1 database upgrades are executed.

Before running the RMS 13.1 Data Migration Tool:

- Make a backup of all your objects and database schema.
- Check that RMS 13.1 is installed.
- Review the RMS 13.1 Release Notes (rms-131-rn.pdf).

Configure RMS Data Migration Tool

1. Change directories to DB_PATCH_DIR/mom-dbpatch/13.1/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. If there is an existing controller.cfg in DB_PATCH_DIR/mom-dbpatch/13.1/rms/ from a previous running of the 13.1 upgrade and its contents are still valid, it can be reused. Otherwise, copy DB_PATCH_DIR/mom-dbpatch/13.1/rms/templates/controller.cfg to DB_PATCH_DIR/mom-dbpatch/13.1/rms/.
   - Open the controller.cfg file you just created and replace the tokens for the following variables with the appropriate values:
     i. Export PATCH_DIR=DB_PATCH_DIR/mom-dbpatch/13.1/<product>
     ii. export SCHEMA_OWNER=<The name of the RMS schema>
     iii. export MMUSER=<The name of the RMS schema>
     iv. export PASSWORD=<password for the MMUSER schema>
     v. export ORACLE_SID=<SID for the database the MMUSER schema resides in>
6. Configure the following files in the DB_PATCH_DIR/mom-dbpatch/13.1/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.

Note: For descriptions of this data, refer to the RMS 13.1 Data Model document (rms-131-dm.pdf).

- **store.dat**

  store.dat is used to update the time zone for a given store. To populate store.dat, use the query

  ```
  select store || '|' || 'EST' from store;
  ```

  And modify the time zone "EST" to represent the correct time zone for each store. You should retain the header "STORE|TIMEZONE_NAME" as it is in the sample store.dat.

- **elc_comp.dat**

  elc_comp.dat is used to update the system generated indicator for comp_id. To populate elc_comp.dat, use the query

  ```
  select comp_id || '|' || 'Y' from elc_comp;
  ```

  You should retain the header "COMP_ID|SYSTEM_GENERATED_IND" as it is in the sample elc_comp.dat.

- **system_options.dat**

  This updates the RMS system options (system_options). Replace the default values in the template system_options.dat file with the correct values for your schema. You should not update the first column.

- **sa_system_options.dat**

  This updates the sales audit system options (sa_system_options). Replace the default values in the template sa_system_options.dat file with the correct values for your schema. You should not update the first column.

- **ce_comp_min_max.dat**

  This inserts a new table "ce_comp_min_max." Replace the values in the template ce_comp_min_max.dat file with the correct values for your schema. You should retain the header "COUNTRY_ID|COMP_ID|MIN_AMT|MAX_AMT|CURRENCY_CODE" as it is in the sample ce_comp_min_max.dat.

7. If the item_supp_country table contains a high volume of rows (i.e. more than 500,000 rows), follow these steps:

   i. Enter the item, supplier, manufacturing country id and primary_manu_country_ind into a spreadsheet.

   ii. Have your tech team load the data manually from the spreadsheet to upg_item_manu_country table. The load strategy will depend on the recommendation of the tech person, but Oracle highly recommends SQL Loader.

   Otherwise, run the following insert statement into your RMS schema manually. You can modify the default values if necessary. If the item_supp_country table contains a high volume of rows, it is not advisable to run this insert query:

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

Run the RMS Data Migration Tool

1. Change directories to DB_PATCH_DIR/mom-dbpatch/13.1/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 are up to date:

   ```
   $ ./rms_controller.ksh PREVALIDATION
   ```

4. Run migration tool.

   ```
   $ ./rms_controller.ksh UPGRADE
   ```

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

   ```
   $ ./rms_controller.ksh CLEANUP
   ```

6. Refer to the files in the "log" and "error" dir for details if there are problems during migration.
7. You will need to rebuild synonyms for any additional RMS users. See the chapter "Set up Additional RMS Users" for instructions on creating these synonyms.

Configure ReIM Data Migration Tool

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

1. Change directories to DB_PATCH_DIR/mom-dbpatch/13.1/reim.
2. If there is an existing controller.cfg in DB_PATCH_DIR/mom-dbpatch/13.1/reim/ from a previous running of the 13.1 upgrade and its contents are still valid, it can be reused. Otherwise, copy DB_PATCH_DIR/mom-dbpatch/13.1/reim/templates/controller.cfg to DB_PATCH_DIR/mom-dbpatch/13.1/reim/
   - Open the controller.cfg file you just created and replace the tokens for the following variables with the appropriate values:
     i. Export PATCH_DIR=DB_PATCH_DIR/mom-dbpatch/13.1/<product>
     ii. export SCHEMA_OWNER=<The name of the RMS schema>
     iii. export MMUSER=<The name of the RMS schema>
     iv. export PASSWORD=<password for the MMUSER schema>
     v. export ORACLE_SID=<SID for the database the MMUSER schema resides in>

Run the RelM Data Migration Tool

1. Change directories to DB_PATCH_DIR/mom-dbpatch/13.1/reim.
2. If rerunning the data migration process, clear the contents of the "processed" directory.
3. Run migration tool.

```
$ ./reim_controller.ksh UPGRADE
```

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

```
$ ./reim_controller.ksh CLEANUP
```

5. Refer to the files in the "log" and "error" dir for details if there are problems during migration.
6. You will need to rebuild synonyms for any additional RMS users. See the chapter "Set up Additional RMS Users" for instructions on creating these synonyms.

A

Appendix: Oracle 11g Database Parameter File

```
##############################################################################
# Oracle 11.1.0.x Parameter file
#
# NOTES: Before using this script:
#        1.  Change <datafile_path>, <admin_path>, <utl_file_path>, <diag_path>
and <hostname>
#            values as appropriate.
#        2.  Replace the word SID with the database name.
#        3.  Size parameters as necessary for development, test, and production
environments.
# ----------------------------------------------------------------------
# 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            = 11.1.0
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
optimizer_features_enable     = 11.1.0.7
optimizer_mode            = CHOOSE    # Oracle Retail required
```

```
plsql_optimize_level                    = 2         # 10g change; use this setting
to optimize plsql performance
processes             = 500       # 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
```

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.

```
spool create_user.log

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
/

grant developer,
      select_catalog_role,
      alter session,
      analyze any,
      create any synonym,
      create any type,
      create database link,
      create library,
      create procedure,
      create public database link,
      create public synonym,
      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
/

spool off

exit;
```

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

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 their 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 install scripts that must be run prior to HTS Upload to populate the following tables. These are one-time installs upon implementation of the product and must be maintained by the client:

- ELC_COMP
- QUOTA_CATEGORY (via the quota_category.sql script)
- OGA (via the oga.sql script)
- COUNTRY_TARIFF_TREATMENT (via the country_tariff_treatment.sql script)
- HTS_CHAPTER (via the hts_headings.sql script)
- TARIFF_TREATMENT (via 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 (via 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 via 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 'Downloaded'; 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

- 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 contains the sum of the Total Duty, Total Taxes, and Total Other values.

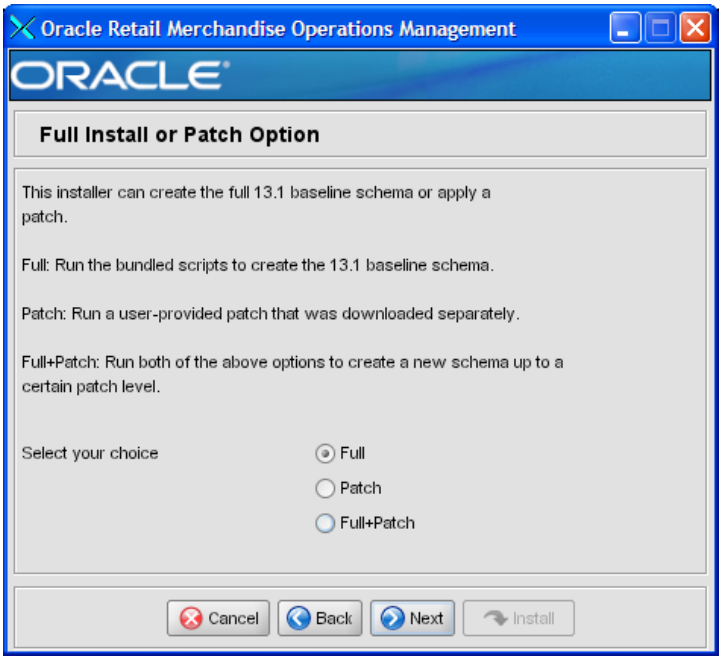
G

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.

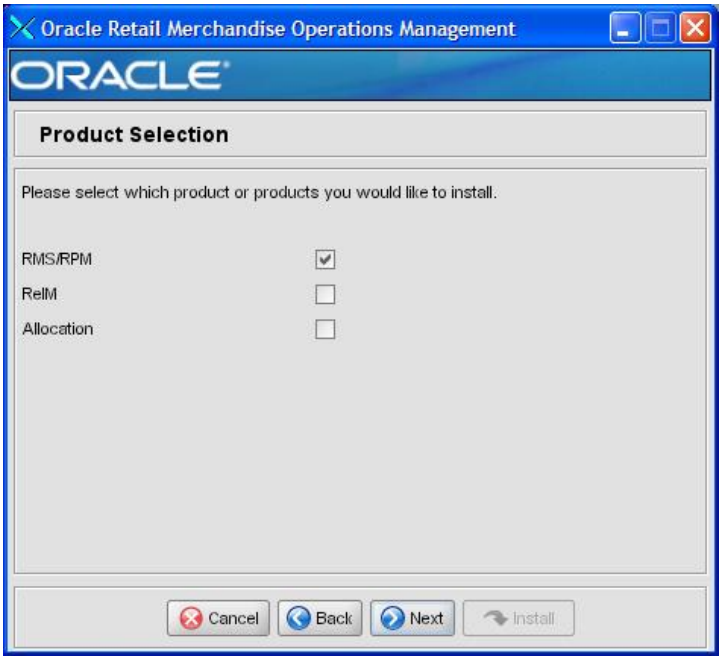
Screen: Full Install or Patch Option



Fields on this Screen:

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 13.1 release, select Full. If installing 13.1 or later, select Patch and the installer prompts for the location of the patch files on the next screen. Note: This installer cannot be used to apply any patches earlier than 13.1. To apply the 13.0.1 or 13.0.2 patch, please use the 13.0.1 Full installer, or apply them manually.
Example	Full

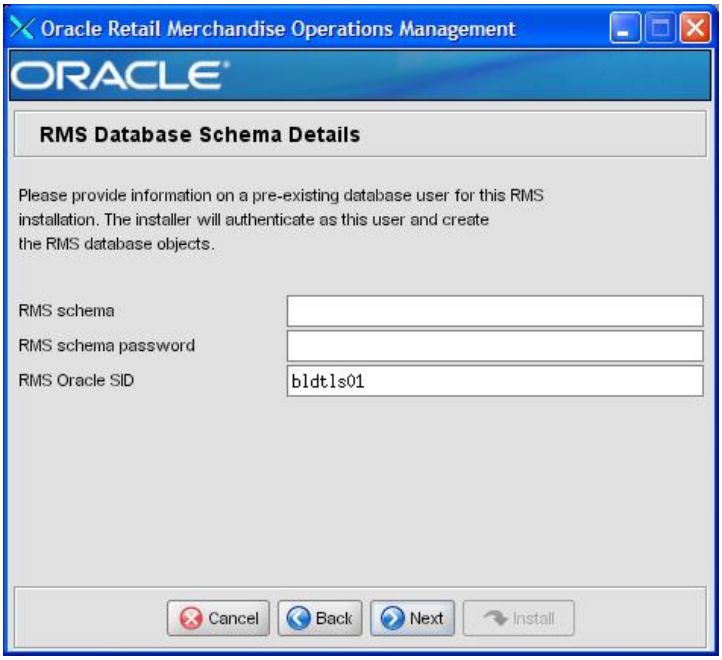
Screen: Product Selection



Fields on this Screen:

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



Fields on this Screen:

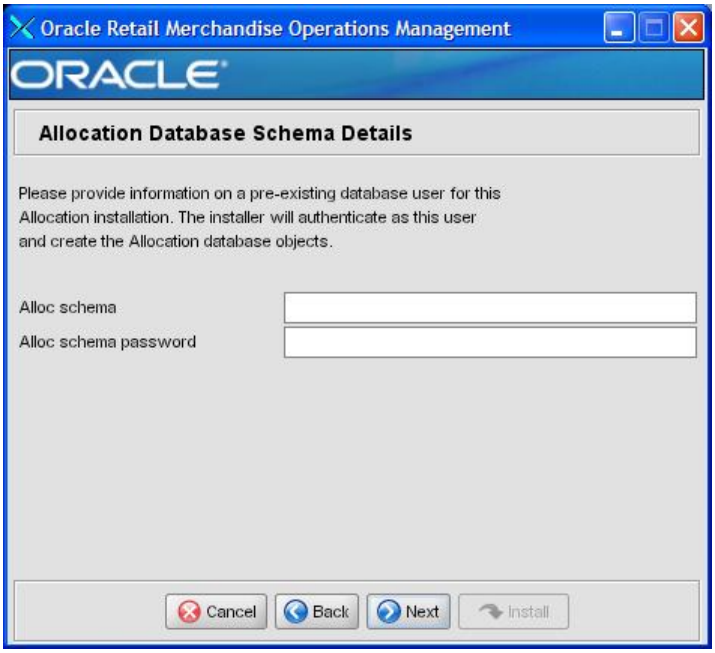
Field Title	RMS schema
Field Description	Provide the RMS database user here. The installer logs into the database as this user to create the RMS schema. This user must already exist in the database when the RMS database schema installer is run.
Example	RMS

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	rmsdb

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

Screen: Allocation Database Schema Details



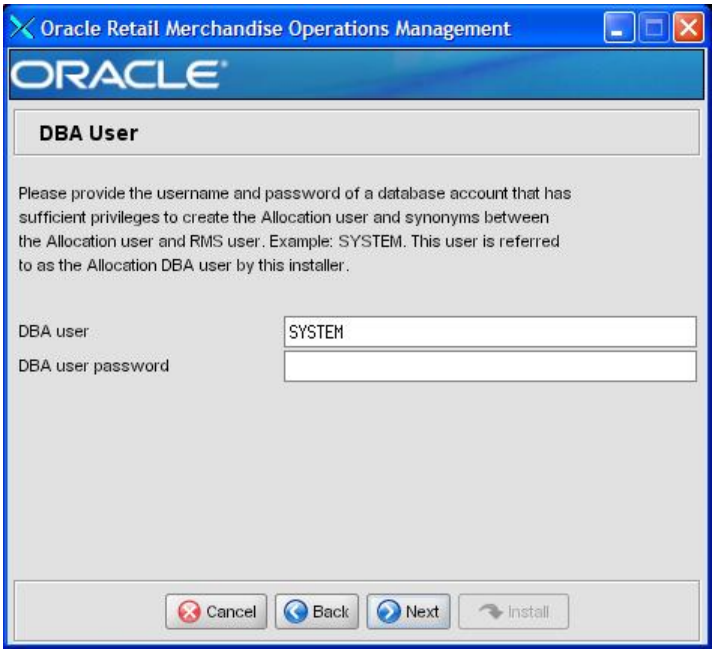
Fields on this Screen:

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	RMS

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



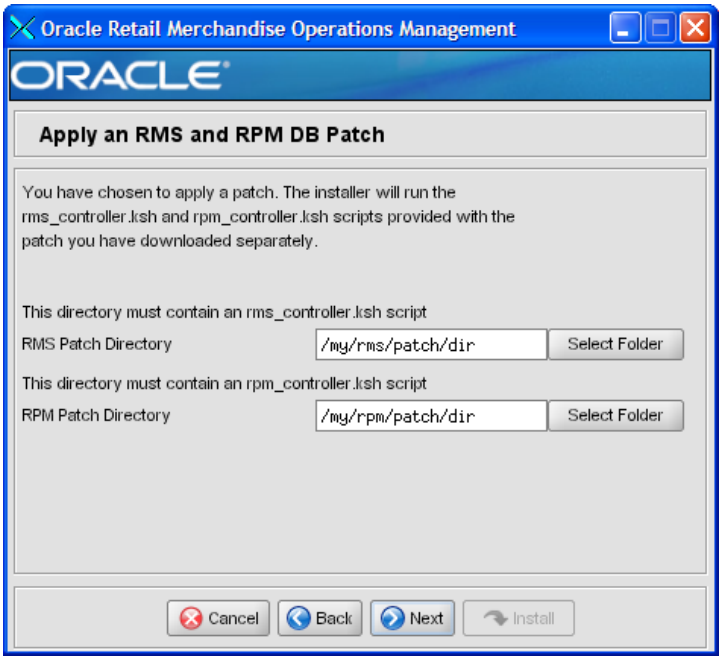
Fields on this Screen:

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

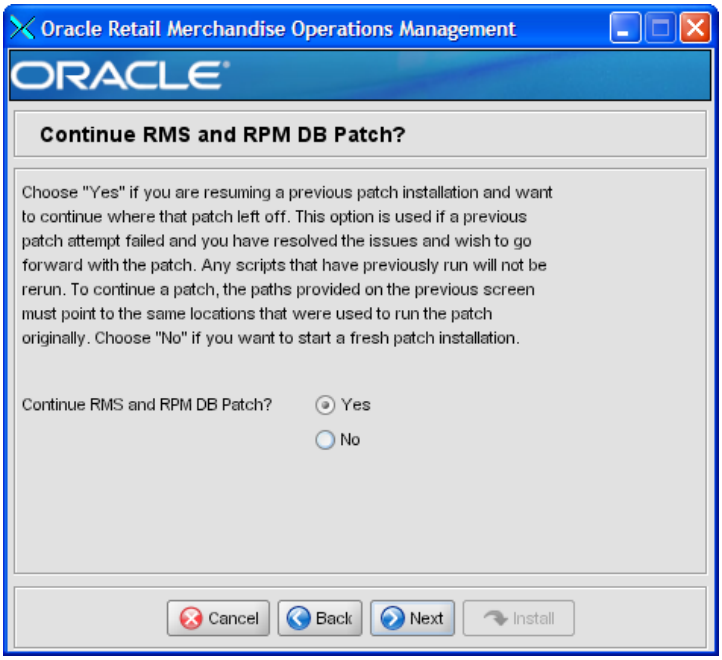


Fields on this Screen:

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 13.1.x patches **Note:** The patch option is intended for patches starting with 13.1.

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 13.1.x patches **Note:** The patch option is intended for patches starting with 13.1.

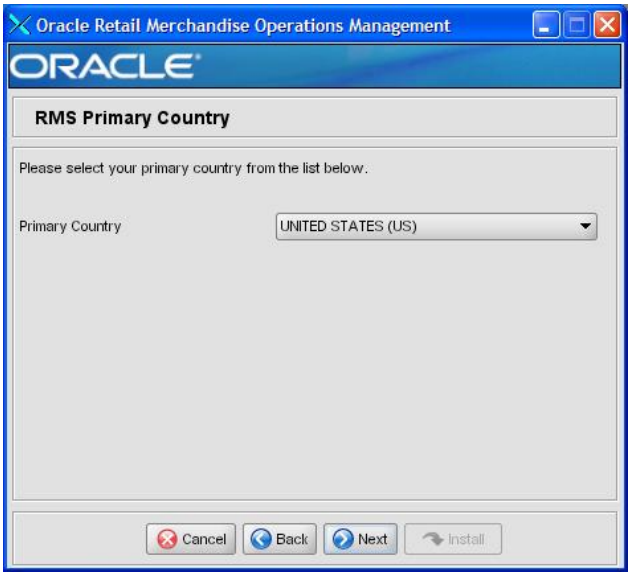
Screen: Continue RMS and RPM DB Patch



Fields on this Screen:

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.

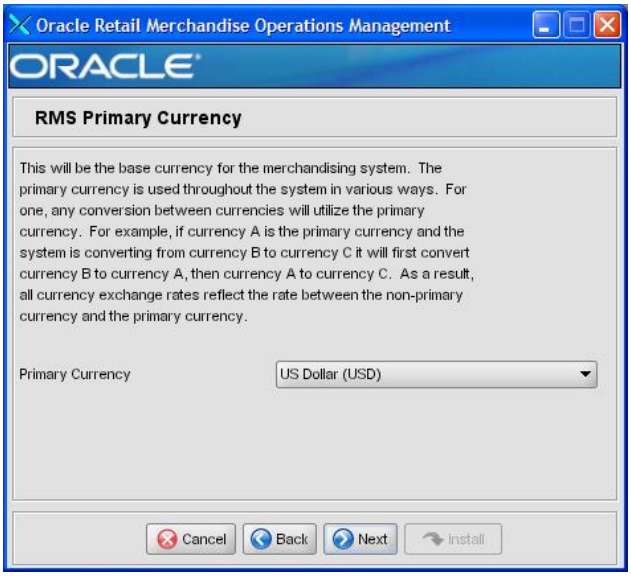
Screen: RMS Primary Country



Fields on this Screen:

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

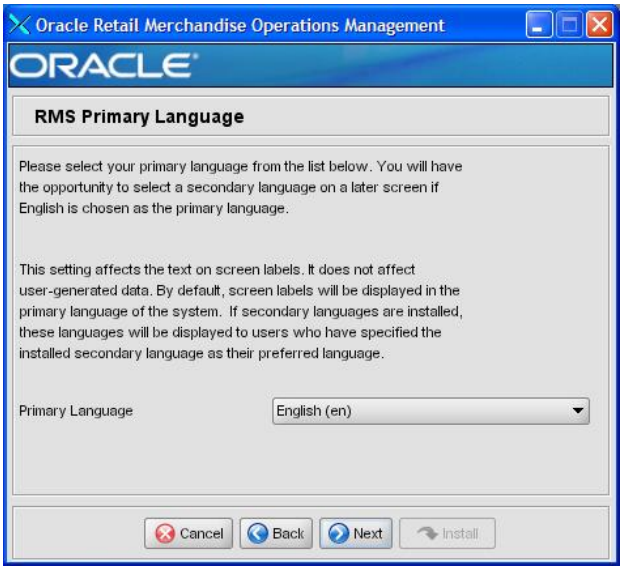
Screen: RMS Primary Currency



Fields on this Screen:

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

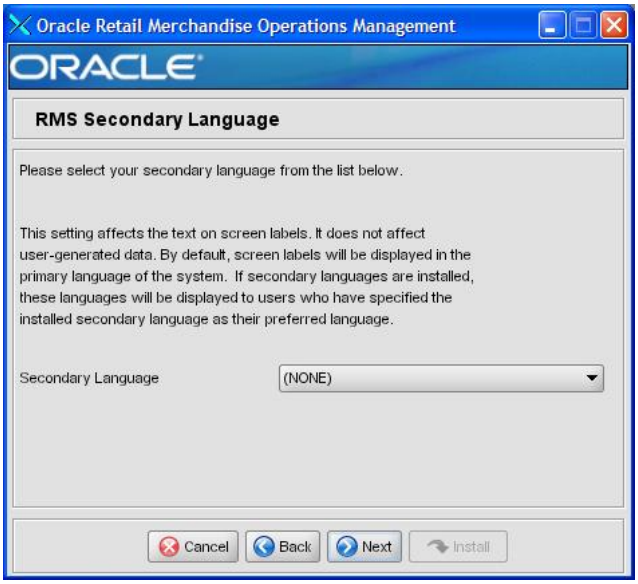
Screen: RMS Primary Language



Fields on this Screen:

Field Title	Primary Language
Field Description	Choose your primary language from the list provided. You have an opportunity to select a secondary language on another screen.
Example	en

Screen: RMS Secondary Language



Fields on this Screen:

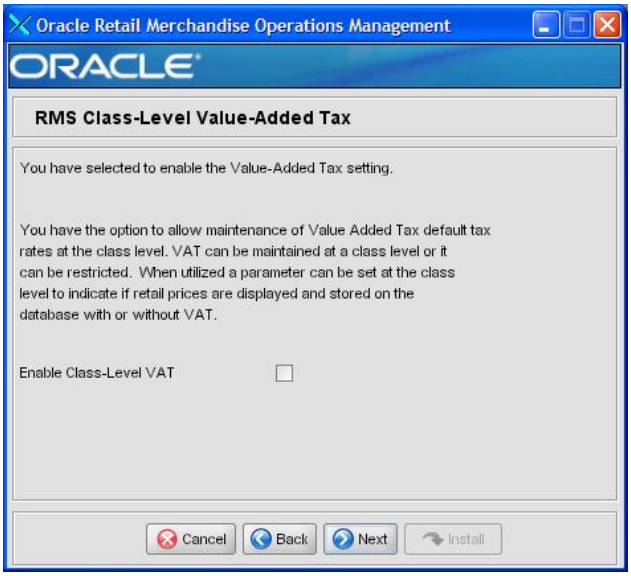
Field Title	Secondary Language
Field Description	This screen only appears if English is selected as the primary language. Choose your secondary language from the list provided. Among the individual language choices are the (NONE) and (ALL) selections, which use no secondary language or all secondary languages, respectively.
Example	none

Screen: RMS Value-Added Tax (VAT)

Fields on this Screen:

Field Title	Enable VAT?
Field Description	Select Yes if you will use VAT.
Example	No

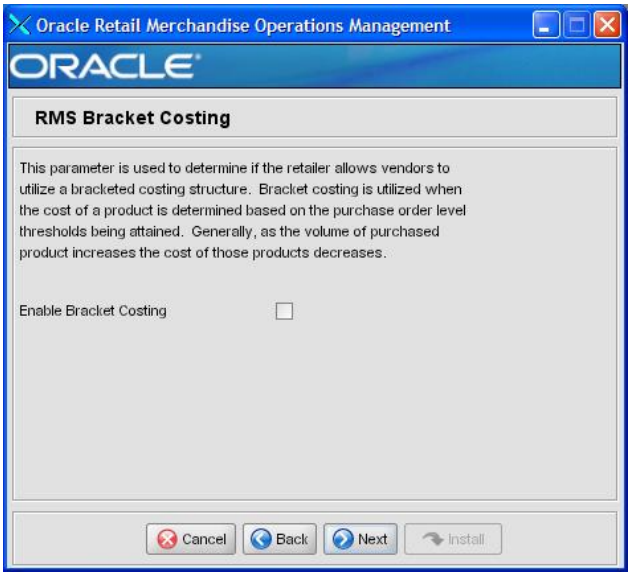
Screen: RMS Class Level Value-Added Tax



Fields on this Screen:

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.
Example	No

Screen: RMS Bracket Costing



Fields on this Screen:

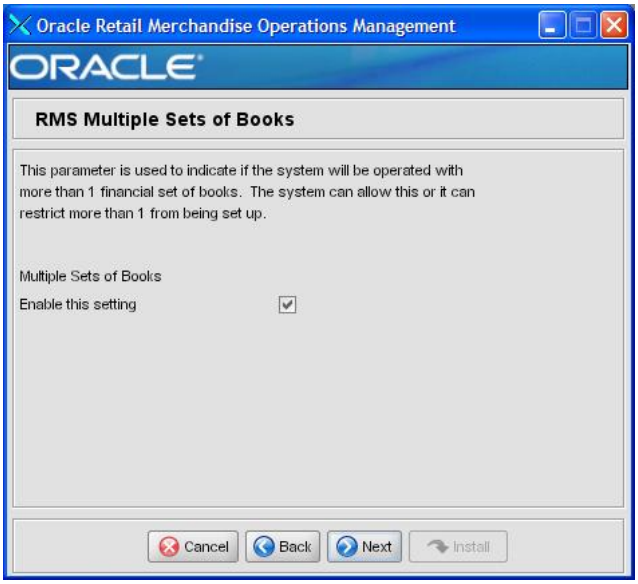
Field Title	Enable Bracket Costing?
Field Description	Select Yes if you allow vendors to use a bracketed costing structure.
Example	Yes

Screen: RMS Wholesale

Fields on this Screen:

Field Title	Enable Wholesale?
Field Description	Wholesale functionality in RMS
Example	Yes

Screen: RMS Multiple Sets of Books



Fields on this Screen:

Field Title	Enable Multiple Sets of Books?
Example	Yes

Screen: RMS Supplier Sites

Fields on this Screen:

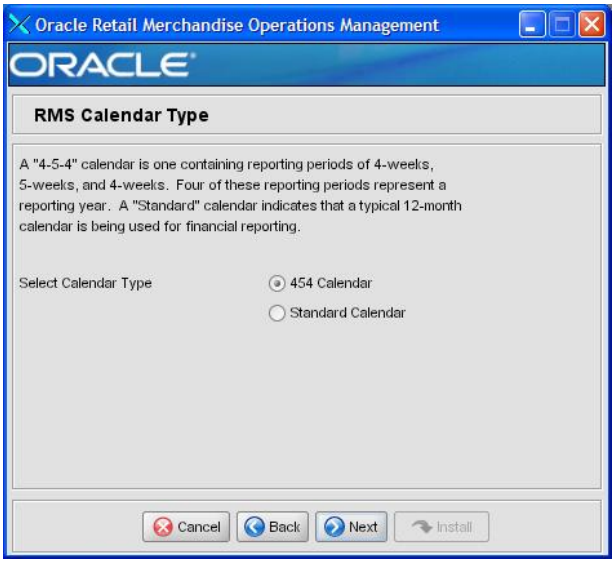
Field Title	Enable Supplier Sites?
Example	Yes

Screen: RMS Freight Terms Loading

Fields on this Screen:

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.

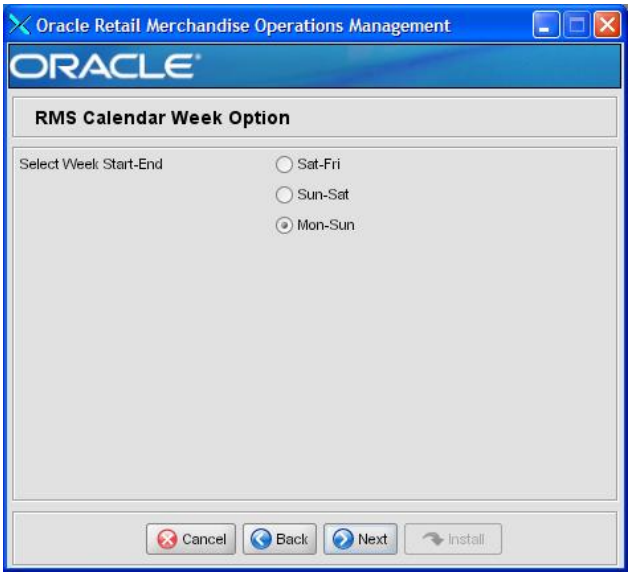
Screen: RMS Calendar Type



Fields on this Screen:

Field Title	Calendar Type
Example	454 Calendar

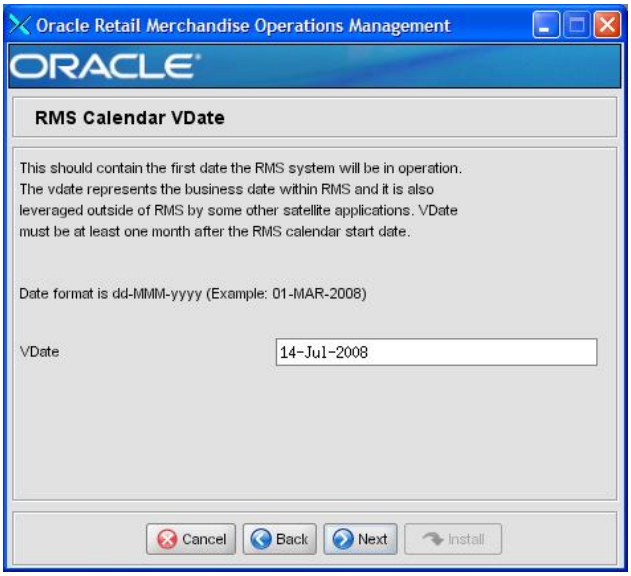
Screen: RMS Calendar Week Option



Fields on this Screen:

Field Title	Week Start-End
Example	Mon-Sun

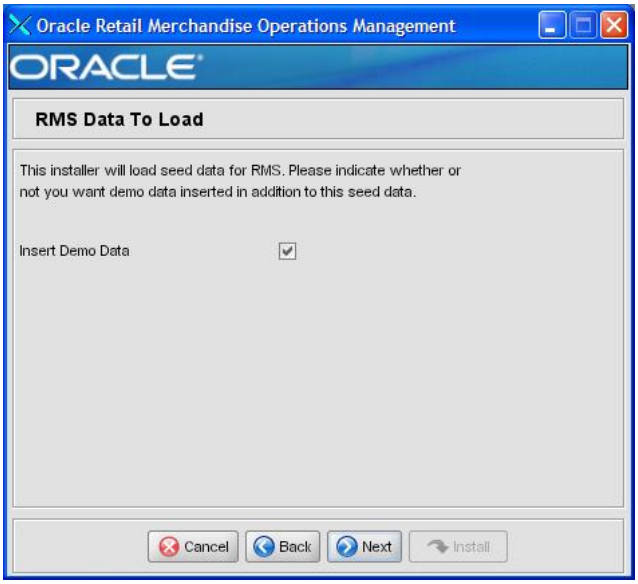
Screen: RMS Calendar VDate



Fields on this Screen:

Field Title	VDate
Example	01-MAR-2001

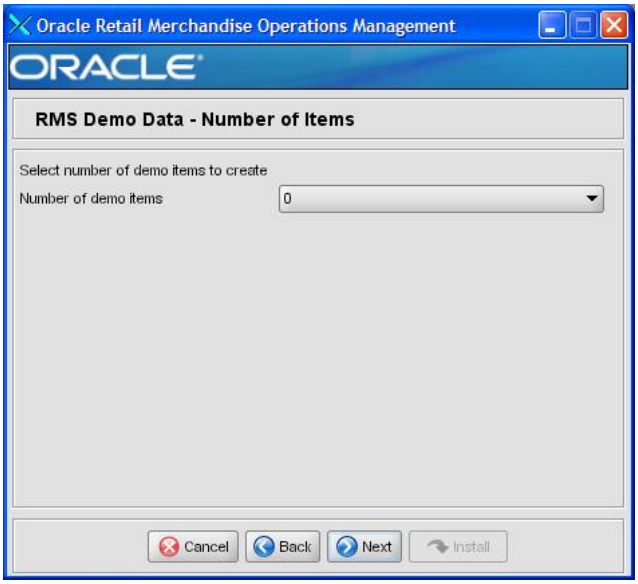
Screen: RMS Data to Load



Fields on this Screen:

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

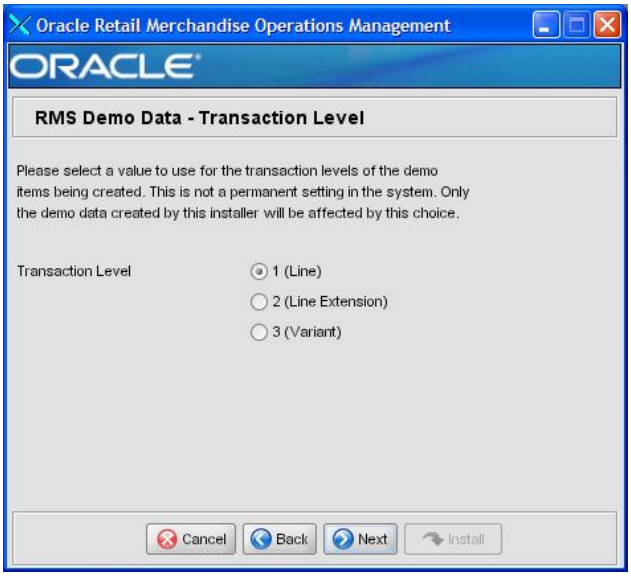
Screen: RMS Demo Data – Number of Items



Fields on this Screen:

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



Fields on this Screen:

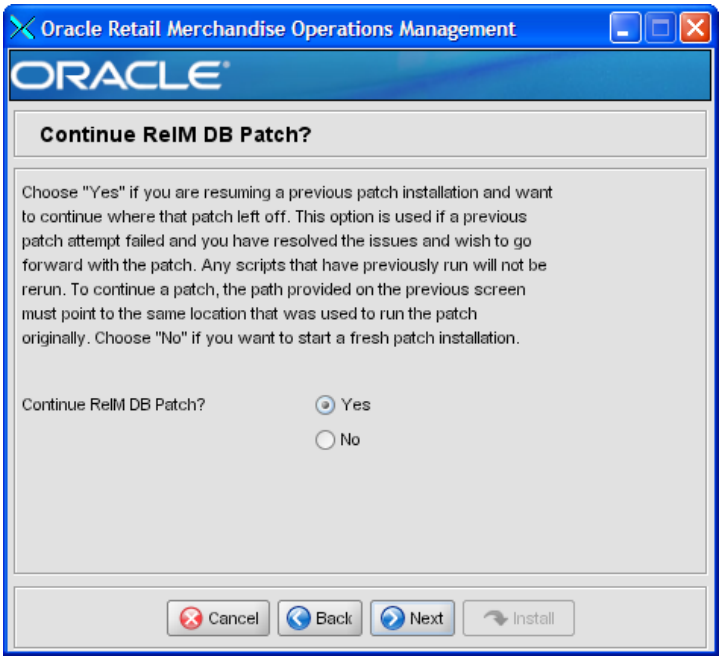
Field Title	Transaction Level
Field Description	If you chose to insert demo items on the previous screen, you are asked to provide a transaction level value for these items.
Example	1

Screen: Apply ReIM DB Patch

Fields on this Screen:

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 13.1.x patches **Note:** The patch option is intended for patches starting with 13.1.

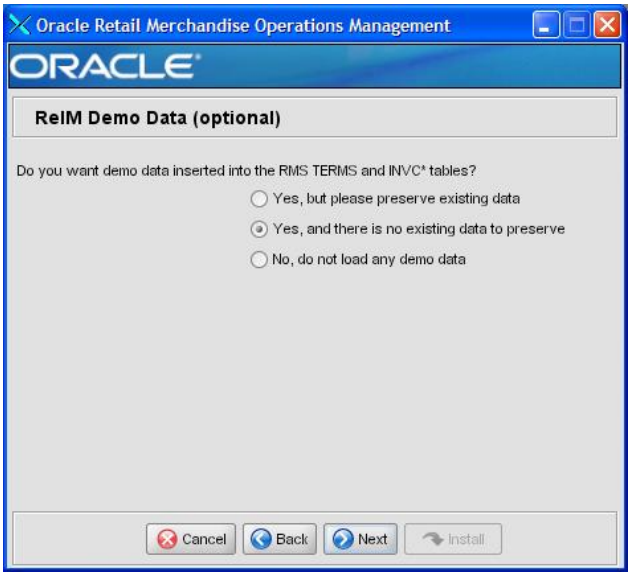
Screen: Continue ReIM DB Patch



Fields on this Screen:

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)



Fields on this Screen:

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.

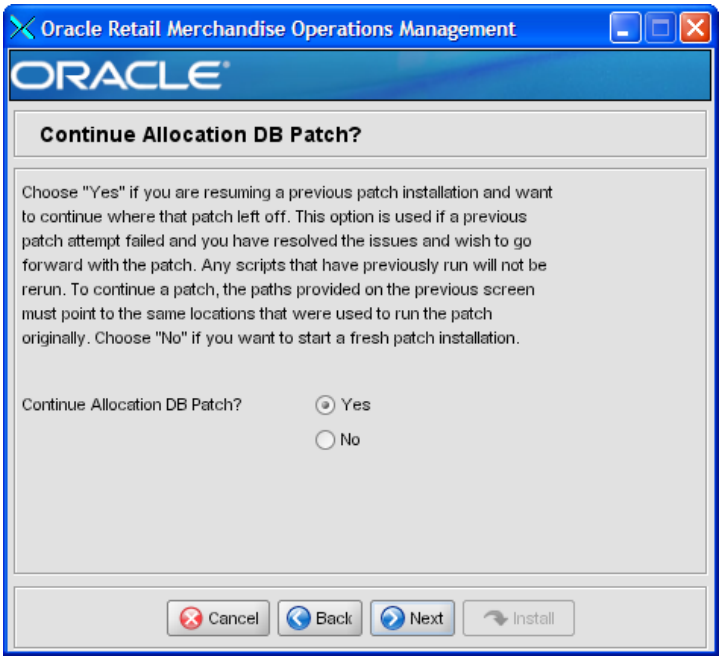
Screen: Apply Allocation DB Patch

Fields on this Screen:

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. The Alloc controller is used to apply the necessary Allocation 13.1 upgrades to the Allocation schema. **Note:** The directory you choose must contain an alloc_controller.ksh file.
Example	/my/alloc/patch/dir for all 13.1.x patches **Note:** The patch option is intended for patches starting with 13.1.

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. The Alloc_RMS controller is used to apply the necessary Allocation 13.1 upgrades to the RMS schema **Note:** The directory you choose must contain an alloc_rms_controller.ksh file.
Example	/my/alloc_rms/patch/dir for all 13.1.x patches **Note:** The patch option is intended for patches starting with 13.1.

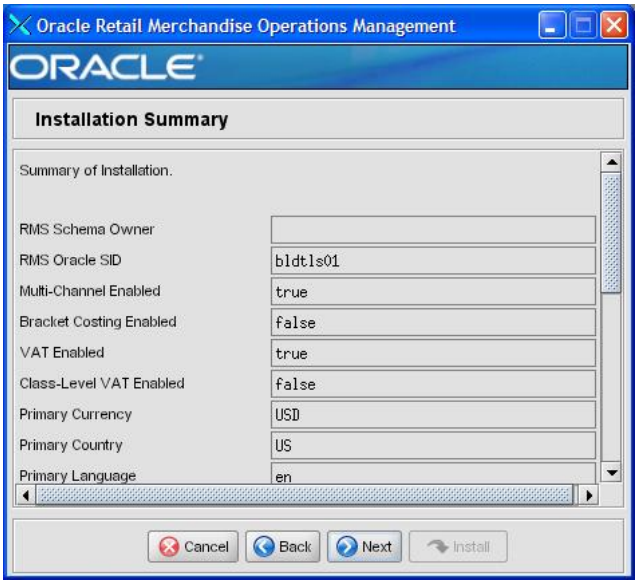
Screen: Continue Allocation DB Patch



Fields on this Screen:

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.

Screen: Summary



Fields on this Screen:

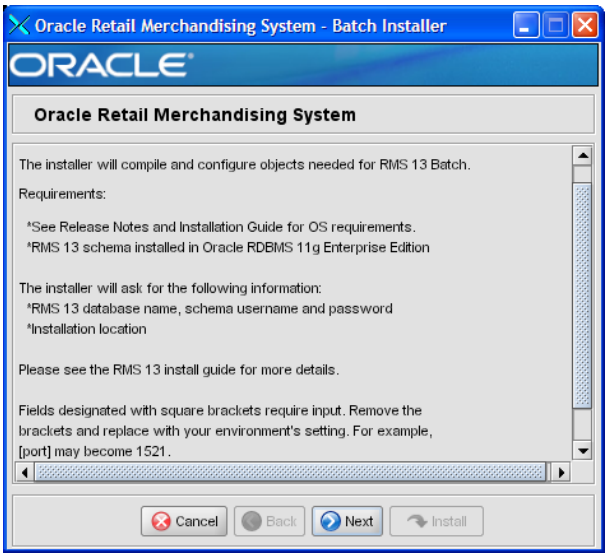
All of the fields on this summary screen are read-only. In GUI mode of the installer, this screen provides the opportunity to review inputs and go back to previous screens to correct them if necessary.

H

Appendix: RMS Batch Installer Screens

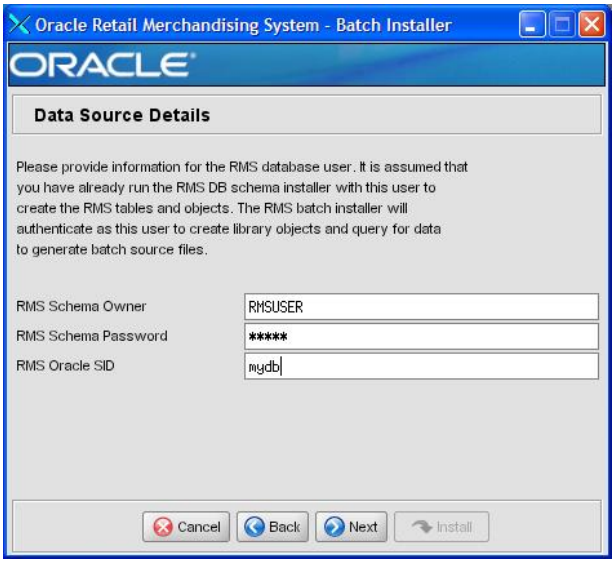
You need the following details about your environment for the installer to successfully compile and install the RMS batch programs. Depending on the options you select, you may not see some screens or fields.

Screen: Welcome



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

Screen: DataSourceDetails



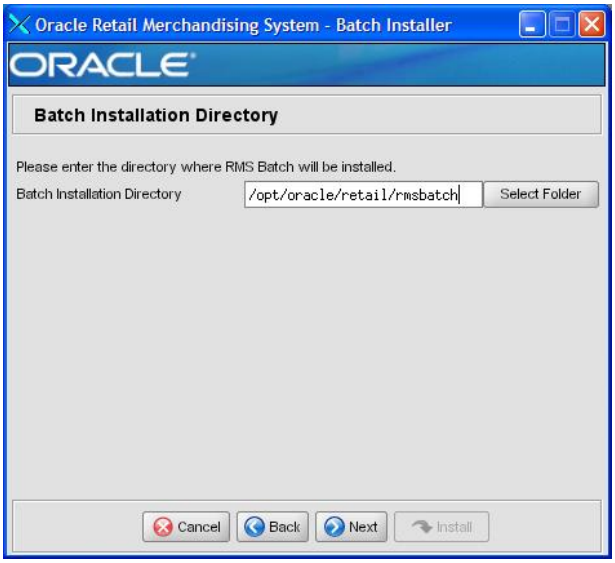
Fields on this Screen:

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	RMSUSER

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	rmsdb

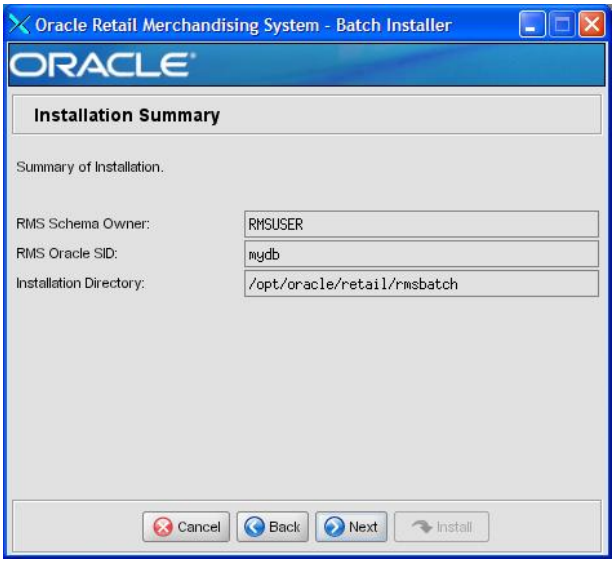
Screen: Batch Installation Directory



Fields on this Screen:

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



Fields on this Screen:

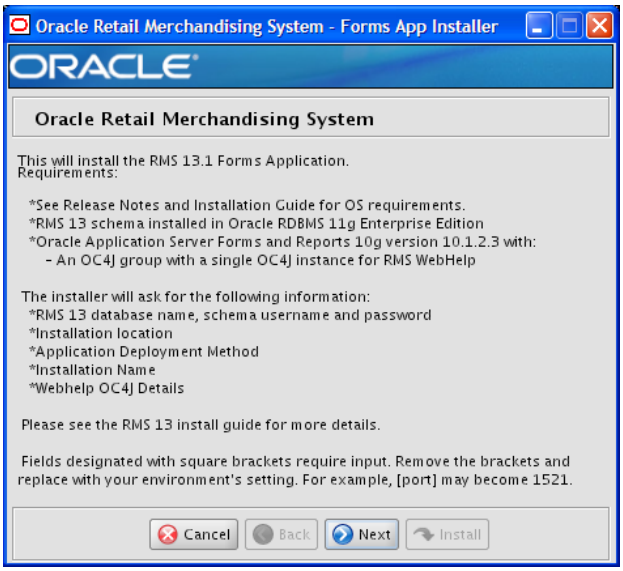
All of the fields on this summary screen are read-only. In GUI mode of the installer, this screen provides the opportunity to review inputs and go back to previous screens to correct them if necessary.

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: Data Source Details

Fields on this Screen:

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

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	Rmsdb

Screen: Application Installation Directory

Fields on this Screen:

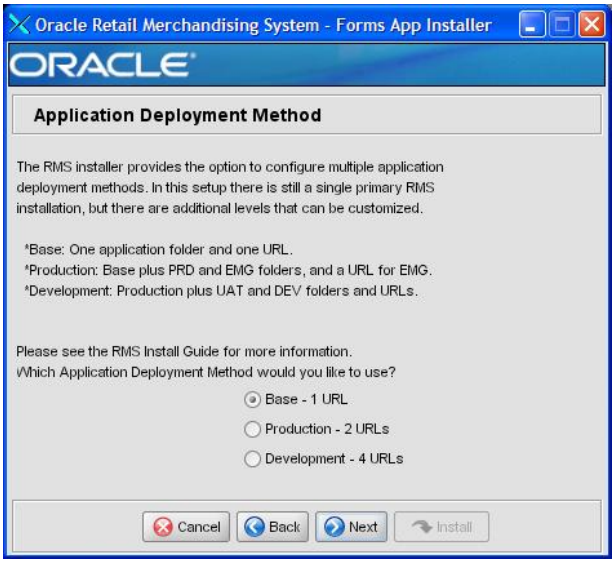
Field Title	Application Installation Directory
Field Description	The location where the RMS Application (toolset, forms and reports) will be installed. The RMS $MMHOME path will be a subdirectory of this directory, named "base".
Example	/u01/oracle/retail

Screen: Installation Name

Fields on this Screen:

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 Oracle Application Server instance.
Example	rms13inst

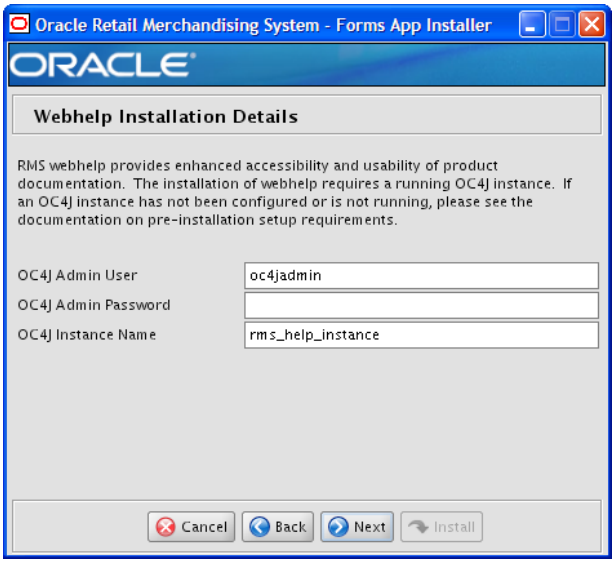
Screen: Application Deployment Method



Fields on this Screen:

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: Webhelp Installation Details



Fields on this Screen:

Field Title	OC4J Admin User
Field Description	Username of the admin user for OC4J instance to which the RMS Webhelp application is being deployed.
Example	oc4jadmin

Field Title	OC4J Admin Password
Field Description	Password for the OC4J admin user. You chose this password when you created the OC4J instance.

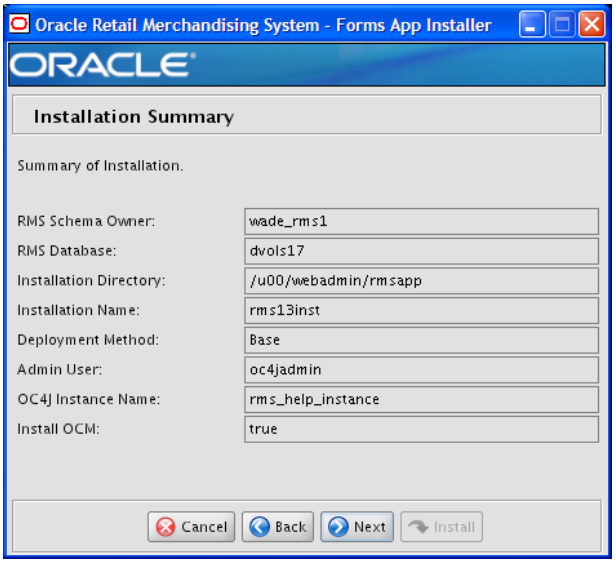
Field Title	OC4J Instance Name
Field Description	Name of the OC4J instance that was created for the RMS Webhelp application.
Example	rms_help_instance

Screen: Install OCM

Fields on this Screen:

Field Title	Install OCM
Field Description	Install OCM checkbox. This field gives you the option to install or not install OCM. The default option is checked. It is highly recommended you install OCM if you meet the requirements.
Example	Checked/True

Screen: Summary



Fields on this Screen:

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. Look for duplicate properties in the ant.install.properties file. Some properties are set on multiple pages to ensure default values when a page is only displayed under certain conditions. For example, if there are two instances of input.property.name, remove all but the last one.
3. Run the installer again with the **silent** argument.

> **Example:** install.sh silent

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.

"Could not create system preferences directory" Warning

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.

"Couldn't find X Input Context" Warnings

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:

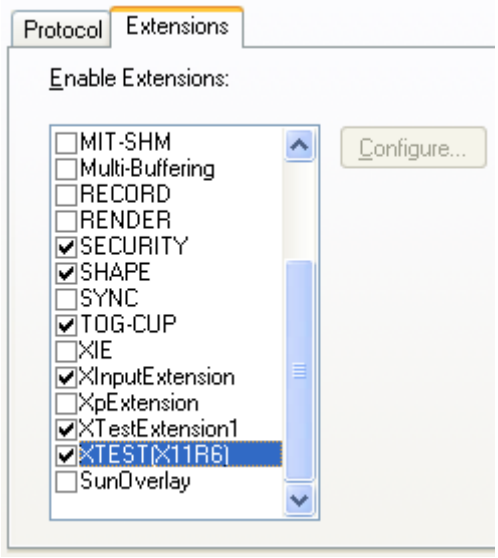
```
XTEST extension not installed on this X server: Error 0
```

Solution:

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

Enabling XTEST in Exceed:

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:



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

Couldn't 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.

Example:

```
frmpcmp.sh userid=$UP module_type=form module=FORM_OR_MENU
```

RIB Errors

At random times, the RIB will get certain errors such as GETNXT(?,?,?,?,?,?,?) and/or ORA-21700 object does not exist or is marked for delete. This is very confusing because you may research and find that the object exists and is valid.

You must re-initialize the reference to reference an existing object. You do this by:

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

"Error Connecting to Database URL"

Symptom:

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

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

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

Solution:

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

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

It may mean that the installer is using the incorrect library path variables for the platform you are installing on. Open the file <STAGING_DIR>/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.

M

Appendix: Manual Application Installation

It is strongly recommend that you use the installer to install the RMS Application. Below are the instructions to manually install the application portion of RMS.

It is assumed that Oracle Application Server 10g version 10.1.2.3 (OAS) 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 10g for manual compilation of RMS 13 forms modules, Oracle Developer Suite 10g Release 2 (10.1.2.3) must be used. Please refer to the Oracle Developer Suite 10g Release 2 documentation for the steps to manually compile objects.

> **Note:** It is necessary to have $ORACLE_HOME/network/admin/tnsnames.ora file configured in this OAS 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 Oracle Application Server 10g (10.1.2.3) has been installed

1. The T2kMotif.rgb file that is sent out with Oracle Application Server 10g (10.1.2.3) must be modified. It located at the following location:

   $ORACLE_HOME/guicommon/tk/admin

   Make a copy of the file ORACLE_HOME/guicommon/tk/admin/Tk2Motif.rgb, and name it Tk2Motif.rgb_ORIG (for example).

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

   ```
   Tk2Motif*fontMapCs: iso8859-2=UTF8
   ```

2. The Logon to the application server as the oretail user,
3. Set the DISPLAY variable to the IP address plus ":0.0" (ie: 10.1.1.1:0.0) of the application server.
4. Set the following variables:

   > **Note:** ORACLE_HOME is the location where Oracle Application Server 10g (10.1.2.3) has been installed

   - 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_HOME/guicommon/tk/admin
- UP=<RMS schema owner>/<RMS schema password>@<RMS database>

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

RMS Toolset Installation

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 pll2plx10gr2_toolset to compile all Toolset .pll's.

   Note: If the pll2plx10gr2_toolset script is not used and the libraries are compiled individually, then they must be compiled in the following order (which is noted in the pll2plx10gr2_toolset 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
4. Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully).
5. Remove all newly created .plx files.
6. Copy all forms (*.fmb files) in the INSTALL_DIR/base/toolset/src directory to the INSTALL_DIR/base/toolset/bin directory.
7. Run fmb2fmx10gr2_fm (in INSTALL_DIR/base/toolset/bin) to compile the Toolset reference forms.
8. Remove all newly created fm_*.fmx files (reference forms should not have executable files).
9. Run fmb2fmx10gr2 (in INSTALL_DIR/base/toolset/bin) to generate Toolset 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/toolset/bin directory.

11. 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
```

12. Copy all menus (*.mmb files) in the INSTALL_DIR/base/toolset/src directory to the INSTALL_DIR/base/toolset/bin directory.
13. Run mmb2mmx10gr2 (in INSTALL_DIR/base/toolset/bin) to generate Toolset runtime menus – .mmx's.
14. 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.

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

RMS Forms Installation

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 pll2plx10gr2_forms to compile all RMS .pll's.
4. 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.
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. Run fmb2fmx10gr2_fm (in INSTALL_DIR/base/rms/forms/bin) to compile the RMS reference forms.
7. Remove all newly created fm_*.fmx files (reference forms should not have executable files).
8. Run fmb2fmx10gr2 (in INSTALL_DIR/base/rms/forms/bin) to generate RMS runtime forms – .fmx's.
9. 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/rms/forms/bin directory.

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

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

11. Copy all menus (*.mmb files) in the INSTALL_DIR/base/rms/forms/src directory to the INSTALL_DIR/base/rms/forms/bin directory.
12. Run mmb2mmx10gr2 (in INSTALL_DIR/base/rms/forms/bin) to generate RMS runtime menus – .mmx's.
13. Check to make sure that each .mmb file has a corresponding .mmx file.
14. Remove all .mmb files from INSTALL_DIR/base/rms/forms/bin.

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

Configure Oracle Application Server 10g for RMS

> **Note:** The proper Oracle Application Server 10g (10.1.2.3) components must be started in order to run Oracle Forms applications.
>
> **Note:** ORACLE_HOME refers to the location where Oracle Application Server 10g (10.1.2.3) Forms and Reports Services is installed.
>
> **Note:** Prior to modifying Oracle Application Server 10g (10.1.2.3) Forms and Reports Services files, a backup of original files should be made.

1. Make a copy of the file ORACLE_HOME/forms/server/default.env, and name it rms.env (for example).
2. Modify the new file rms.env by appending the location of the RMS toolset and forms modules to the FORMS_PATH variable setting, and by adding the NLS_DATE_FORMAT and NLS_LANG variables to the end of this file. The variable FORMS_REJECT_GO_DISABLED_ITEM=FALSE must also be added to rms.env due to changes between Oracle Forms 6i and Oracle Forms 10g Additionally, you must make sure the variable FORMS_USERNAME_CASESENSITIVE=1 exists and is set properly.

> **Example:**
> FORMS_PATH=/u00/rms/toolset/bin:/u00/rms/forms/bi
> n:/u00/oracle/AS10GR2/forms
>
> NLS_DATE_FORMAT=DD-MON-RR
>
> NLS_LANG=AMERICAN_AMERICA.UTF8
>
> FORMS_REJECT_GO_DISABLED_ITEM=FALSE
>
> FORMS_USERNAME_CASESENSITIVE=1

3. Make an entry in the file ORACLE_HOME/network/admin/tnsnames.ora for the Oracle 10g database that was created in Chapter2 (where the RMS 13 schema resides). Appendix C contains a sample tnsnames.ora file entry for an Oracle 10g

database; refer to the sample or following example for a proper entry in file ORACLE_HOME/network/admin/tnsnames.ora.

4. Log into sqlplus as the RMS 13 schema owner (RMS13DEV) and update the lang table so that WEBHELP_SERVER, REPORTS_SERVER, WEBREPORTS_SERVER, and APP_SERVER are correct:
   - WEBHELP_SERVER is the url http://<server>:<port> where <server> is the name or IP address of the server where Oracle AS 10g is installed and <port> is the "Listen" value in ORACLE_HOME/Apache/Apache/conf/httpd.conf
   - REPORTS_SERVER is the value of the reports server created in step 3 above
   - WEBREPORTS_SERVER is reports/rwservlet
   - APP_SERVER is the url http://<server>:<port>/ where <server> is the name or IP address of the server where Oracle AS 10g is installed and <port> is the "Listen" value in ORACLE_HOME/Apache/Apache/conf httpd.conf

   Example: SQL> update lang set
   WEBHELP_SERVER='http://server:7778' where lang=1;

   SQL> update lang set
   REPORTS_SERVER=REP_<SERVER_NAME> where
   lang=1;

   SQL> update lang set
   WEBREPORTS_SERVER='reports/rwservlet' where lang=1';

   SQL> update lang set APP_SERVER='http://server:7778/'
   where lang=1;

5. Modify the file formsweb.cfg located at ORACLE_HOME/forms/server. Create the RMS environment section at the end of this file. Brackets ([ ] in the example below) distinguish a separate environment in this file. Variables to be set in the RMS environment section of formsweb.cfg are: envfile (from step 2 above); width, height, and separateFrame applet parameters; and starting form for the RMS application.

   Example: [rms]

   envfile=rms.env
   width=850
   height=585
   separateFrame=true
   form=rtkstrt.fmx

   If Oracle Single Sign-On is to be used with RMS, then
   - set ssoMode to true.
   - If Resource Access Descriptors are allowed to be dynamically created, then set ssoDynamicResourceCreate to true.

Example: [rms]

envfile=rms.env
width=850
height=585
separateFrame=true
form=rtkstrt.fmx
ssoMode=true
ssoDynamicResourceCreate=true

Comment out the lines with the following variables in formsweb.cfg if they exist:

- baseHTMLjinitiator
- baseHTMLjpi
- jpi_download_page
- jpi_classid
- jpi_codebase
- jpi_mimetype

Example:

```
## baseHTMLjinitiator=basejini.htm
## baseHTMLjpi=basejpi.htm
##
jpi_download_page=http://java.sun.com/products/archive
/j2se/1.4.2_06/index.html
## jpi_classid=clsid:CAFEEFAC-0014-0002-0006-
ABCDEFFEDCBA
##
jpi_codebase=http://java.sun.com/products/plugin/autodl
/jinstall-1_4_2-windows-i586.cab#Version=1,4,2,06
## jpi_mimetype=application/x-java-applet;jpi-
version=1.4.2_06
```

6. Modify the file ORACLE_HOME/forms/java/oracle/forms/registry/Registry.dat by setting default.icons.iconpath to /web_gif/.

   Example: default.icons.iconpath=/web_gif/

7. If NLS_LANG is NOT set in the ORACLE_HOME/forms/server/rms.env then copy the RMS keyboard-mapping file

   INSTALL_DIR/base/sample_files/fmrweb.res to
   ORACLE_HOME/forms/admin/resource/US

   If NLS_LANG is set in the ORACLE_HOME/forms/server/rms.env file then copy the RMS keyboard-mapping file

   INSTALL_DIR/base/sample_files/fmrweb_utf8.res to
   ORACLE_HOME/forms/admin/resource/US

8. Copy the sample file INSTALL_DIR/base/sample_files/rms13unix.conf to ORACLE_HOME/Apache/Apache/conf. rms13unix.conf contains the RMS-specific http listener settings that need to be added to the httpd configuration file that was generated during the installation of AS 10gR2

9. In rms13unix.conf, replace all occurrences of INSTALL_DIR with environment information. The four Apache listener aliases that need to be modified are: /java/help/, /web_gif/, /english/, and /temp/.
10. Add the contents of rms13unix.conf to the end of httpd.conf, or add an include directive in httpd.conf to rms13unix.conf.
11. Reload the Oracle HTTP Server through Oracle Enterprise Manager (OEM) for the new listener settings to take effect. The OEM URL was presented in the End of Installation window at the conclusion of the Oracle AS 10gR2 Forms and Reports Services installation. The default OEM URL should be http://server:1810.
12. Load RMS in Forms 10gR2 mode by entering the following URL in a browser. Prior to testing, the Sun JRE 1.4.1+ plug-in needs to be installed on the client machine. The plug-in can be downloaded from http://java.sun.com/.
    - http://<server>:<port>/forms/frmservlet?config=<env>
    - server = name or IP address of server where Oracle AS 10gR2 is running
    - port = Value of the "Listen" setting in AS10G_ORACLE_HOME/Apache/Apache/conf httpd.conf (default value is 7778)
    - env = name of the environment in brackets in formsweb.cfg (from step 7 above).

Example: http://server:7778/forms/frmservlet?config=rms

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

Note: The first time RMS is accessed, the user is prompted with the following security warning. Click Yes.

If Single Sign-On is not used, or if a Resource Access Descriptor has not been set up for RMS for this user and ssoDynamicResourceCreate is true, then the RMS logon form appears. Enter the appropriateUsername/Password@Connect String information in the corresponding fields:

- Username = RMS Schema Owner or additional Oracle user created
- Password = Username password
- Connect String = Oracle database created in Ch. 1

Example: Username: RMS13DEV
Password: retek
Connect String: prod_db1

13. On the RMS logon form, enter the appropriateUsername/Password@Connect String information in the corresponding fields:
    - Username = RMS Schema Owner or additional Oracle user created
    - Password = Username password
    - Connect String = Oracle database created in Ch. 1

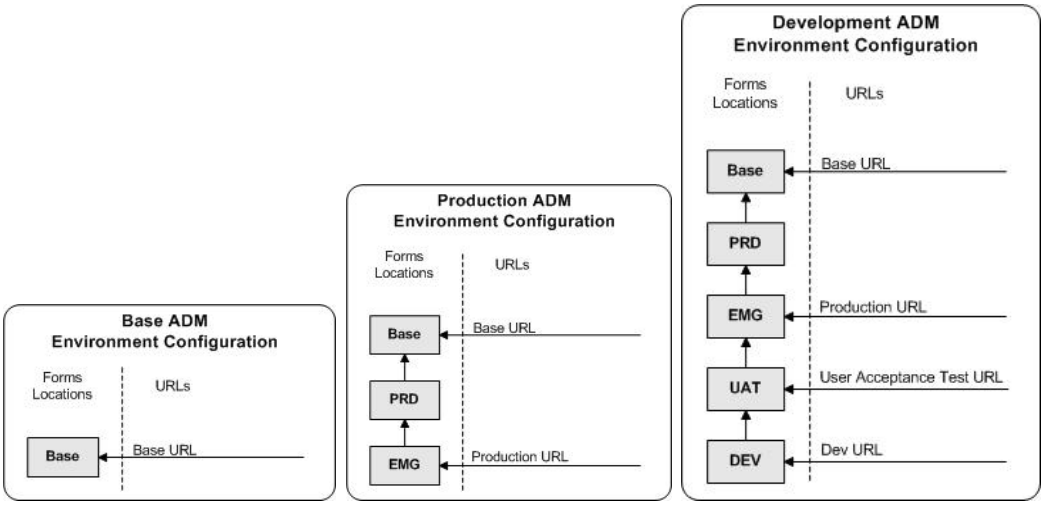
    Example: Username: RMS13DEV
    Password: retek
    Connect String: prod_db1
14. Create an OC4J instance in the 10.1.2 OAS instance running the RMS Application to contain RMS Webhelp.
15. Deploy the RMS help ear file to the OC4J group using the Enterprise Manager web interface. The ear file is located at <STAGING_DIR>/rms/application/rms13/online-help/rms-help.ear. When deploying the ear file, you should provide the application name and context root "rms-help." Using anything other than rms-help will prevent the rms application from accessing the help files.

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 Installation

It is strongly recommend that you use the installer to install the RMS Application. Below are the instructions to manually install the batch portion 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

1. As the oretail user, change directories to INSTALL_DIR/base
2. Set the following variables:

> **Note:**
>
> INSTALL_DIR is the location where RMS 13 will be installed.
>
> Make sure the path for make, makedepend, and the compiler are in $PATH environment variable.

- MMHOME=INSTALL_DIR
- MMUSER=RMS Schema Owner
- PASSWORD=RMS Schema Owner Password
- ORACLE_HOME=Location of Oracle install
- ORACLE_SID=The Oracle Sid for the RMS database

Configure Make File

1. As the oretail user, change directories to INSTALL_DIR/oracle/lib/src
2. 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

3. 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

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.

Re-Validate RMS Database Objects

There is an invalid object compilation script included with the RMS database schema installer package (rms13dbschema.zip).

1. As the oretail user, change directories to the staging directory used for the database schema installer. 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

1. As the oretail user, change directories to INSTALL_DIR/oracle/lib/src
2. To make library dependencies run this command.

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

   Check the libdpnd.log file for errors
3. To make batch libraries

   ```
   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

1. As the oretail user, change directories to INSTALL_DIR/oracle/proc/src
2. To make dependencies run the following command:

   ```
   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.

   ```
   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: 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. Documentation on this may be found in the My Oracle Support document; Doc ID: 244526.1.

Q

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. The following changes in **bold** 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)
```

R

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)
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 jndi_provider.xml file.

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

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

12. Oracle Retail Predictive Application Server (RPAS)
13. Oracle Retail Merchandise Financial Planning (MFP)
14. Oracle Retail Size Profile Optimization (SPO)
15. Oracle Retail Assortment Planning (AP)
16. Oracle Retail Item Planning (IP)
17. Oracle Retail Item Planning configured for COE (IPCOE)
18. Oracle Retail Advanced Inventory Planning (AIP)
19. Oracle Retail Integration Bus (RIB)
20. Oracle Retail Point-of-Service (ORPOS)

21. Oracle Retail Mobile Point-of-Service (ORMPOS)
22. Oracle Retail Analytics Applications
23. Oracle Retail Data Warehouse (RDW)
24. Oracle Retail Workspace (ORW)