Oracle® Retail Merchandising System

Installation Guide
Release 12.1

October 2008

ORACLE

Oracle® Retail Merchandising System Installation Guide, Release 12.1

Primary Author: Paul Kehler

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 and Oracle Retail Demand Forecasting 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 Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") 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.

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

For more information, see the following documents in the Oracle Retail Merchandising System Release 12.1 documentation set:

- *Oracle Retail Merchandising System Data Model*
- *Oracle Retail Merchandising System Release Notes*
- *Oracle Retail Merchandising System Online Help*
- *Oracle Retail Merchandising System Operations Guide – Volume 1*
- *Oracle Retail Merchandising System Operations Guide – Volume 2*
- *Oracle Retail Merchandising System Operations Guide – Volume 3*
- *Oracle Retail Merchandising System Reports User Guide*
- *Oracle Retail Merchandising System User Guide*
- *Oracle Retail Merchandising Batch Schedule*
- *Oracle Retail Merchandising Data Conversion Operations Guide*
- *Oracle Retail Merchandising Implementation Guide*
- *Oracle Retail Sales Audit User Guide*
- *Oracle Retail Trade Management User Guide*

Customer Support

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 Patch Documentation

If you are installing the application for the first time, you install either a base release (for example, 13.0) or a later patch release (for example, 13.0.2). 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. Patch 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

Check Database Server Requirements

General Requirements for a database server running RMS include:

Supported on:	Versions Supported:
Database Server OS	OS certified with Oracle RDBMS 10g Enterprise Edition. Options are: ▪ Oracle Enteprise Linux 4 Patch 5 for Linux x86-64 ▪ AIX5.3
Database Server	Oracle Database 10g Release 2 Enterprise Edition (minimum 10.2.0.3 patchset required) with the following patches and components: Patches: ▪ 4516865 (WRONG PERMISSIONS AFTER INSTALLATION IN OH AND SUBSEQUENT DIRECTORIES) Components: ▪ Oracle Database 10g ▪ Oracle Partitioning ▪ Oracle Net Services ▪ Oracle Call Interface (OCI) ▪ Oracle Programmer ▪ Oracle XML Development Kit ANSI compliant C compiler (certified with OS and database version) Perl compiler 5.0 or later x-Windows interface

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 10.1.2.2. Options are: ▪ Oracle Enteprise Linux 4 Patch 5 for Linux x86-64 ▪ AIX5.3
Application Server	Oracle Application Server Forms and Reports 10g version 10.1.2.2

Check Web Browser and Client Requirements

General requirements for client running RMS include:

Product	Version
JRE Plug-in	Sun JRE Plug-in 1.4.1+
Operating System	Windows 2000 or XP
Processor	Pentium processor
Display resolution	1024x768 resolution
PC Configuration	minimum 256 MB RAM, 450 MHz
Internet Explorer	5.5, 6.0 and higher

Supported Oracle Retail Products

Product	Version
Oracle Retail Price Management (RPM)	12.1
Oracle Retail Allocation	12.1
Oracle Retail Invoice Matching (ReIM)	12.1

Supported Oracle Retail Integration Technologies

Integration Technology	Version
Oracle Retail Integration Bus (RIB)	12.0.1 + hot fixes 1-23

Supported Oracle Applications

Requirement	Version
Oracle E-Business Suite	11.5.10 or 12.0.2

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

Create Staging Directory for RMS Database Files

1. Log into the database server as oretail.
2. Create a staging directory for the RMS database installation software. There should be a minimum of 100 MB disk space available in this location.
3. Copy the rms121dbserver.zip file from the CD/dbserverunix directory to the staging directory. This will be referred to as INSTALL_DIR when installing database software.

4. Change directories to INSTALL_DIR and extract the rms121dbserver.zip file.

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 rms121appserver.zip from the CD/appserverunix directory to staging directory. This will be referred to as INSTALL_DIR when installing application software.
4. Change directories to INSTALL_DIR and extract the file rms121appserver.zip.
5. Make sure all scripts in INSTALL_DIR/forms10gr2_scripts have at least execute permissions for the oretail user and its group (r-xr-x---).

2

Database Installation Tasks

Establish Database Partitioning Strategy

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

Requirements for mandatory and optional partitioning are defined in the Microsoft Excel spreadsheet located in INSTALL_DIR/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 (INSTALL_DIR/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 INSTALL_DIR/ddl/rms12_part.tab or by implementing the process defined below. This file will be used later in the installation process.

> **Note:** Refer to Oracle10g Database Concepts Release 2 Chapter 18 "Partition Tables and Indexes" 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 will update 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 will calculate fewer hash partitions. The typical factor value ranges from 2 to 4 and in special cases, it can be 10 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 will ensure that each partition key has a separate partition and that none are empty. The "Number of Partitions" column will be 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 INSTALL_DIR/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 INSTALL_DIR/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 will contain all data less than 01012004. The second partition will contain all data greater than or equal to 01012004 and less than 01012005. A third "MAXVALUE" partition will automatically be created for all data greater than or equal to 01012005.

When using LIST partitioning, the data definition files will 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 INSTALL_DIR/ddl/part/partition.ksh at the UNIX command prompt. This script will read configuration information from the partition_attributes.cfg file and generate the partitioned DDL file INSTALL_DIR/ddl/rms12_part.tab. This file will be used later during the installation process.

Sample output from partition.ksh:

```
<INSTALL_DIR>/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 ../rms120.tab.
# The partitioned DDL file that will be generated is ../rms120_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
../rms12_part.tab file.
Completed successfully
```

Create the RMS Database

It is assumed that Oracle 10g release 2, with appropriate patches, has already been installed. If not, refer to "*Check Database Server Requirements*" in Chapter 1, "Pre-Installation Tasks" before proceeding**.** Additionally, *INSTALL_DIR* in this section refers to the directory created in "Create Staging Directory for RMS Database Files", Chapter 1.

A partitioning strategy is mandatory for the successful installation of RMS. If this has not been completed, review "Establish Partition Strategy" before continuing.

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

Note: When running the scripts in this section, the following errors may be encountered "ORA-04043 object XXXX does not exist", "ORA-01432 public synonym to be dropped does not exist", "ORA-00942 table or view does not exist", "ORA-29833 indextype does not exist", "ORA-29807 specified operator does not exist", "ORA-29931 specified association does not exist", and "ORA-29816 object being disassociated is not present". These errors can be ignored. The ORA errors are caused by dropping the objects the script is about to create.

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. Copy INSTALL_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.
5. Create a symbolic link from $ORACLE_HOME/pfile/init${ORACLE_SID}.ora to $ORACLE_HOME/dbs/init${ORACLE_SID}.ora.
6. Modify the INSTALL_DIR/create_db/crdb1.sql file. Refer to comments in this file regarding modifications that need to be made.
7. Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/crdb1.sql. Review crdb1.log for errors and correct as needed.
8. Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/crdb2.sql. Review crdb2.log for errors and correct as needed.
9. Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/crdb3.sql. Review JServer.log, context.log and xdb_protocol.log for errors and correct as needed.
10. 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 INSTALL_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 will determine 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 will result in "insufficient space" errors which will require a complete re-install of RMS.

The INSTALL_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

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

Create the Schema Owner

Create an Oracle schema that will own the RMS application. Refer to Appendix D and the section below.

The following grants need to be added for RMS 12.1:

```
grant select on dba_sys_privs to public with grant option;
grant execute on dbms_crypto to public;
```

1. Change directories to INSTALL_DIR/utility
2. The create_user script relies on an empty role, developer and cc_access, being created. Log into sqlplus as sysdba and run the following command to create that role.
   - SQL> create role developer;
   - SQL> create role cc_access;
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 RMS121DEV
     - Password – the password for RMS121DEV
     - Temp Tablespace – the temporary tablespace for RMS121DEV

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 RIB Objects

A partitioning strategy is mandatory for the successful installation of RMS. If this has not been completed, review, "Establish Database Partitioning Strategy" before continuing.

> **Note:** When running the scripts in this section the following errors may be encountered "ORA-04043 object XXXX does not exist" and "Warning: Type created with compilation errors". These errors can be ignored. The ORA errors are caused by dropping the item the script is about to create and the warnings are caused by dependencies on objects that get created later. The warnings will be cleared when objects are revalidated later in the install.

Create RIB Tables and Types

1. Change directories to INSTALL_DIR/rib_objects/ddl.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rib121ddl.sql
   ```
3. Check the log file rib121ddl.log for any errors noting that ORA-04043 errors and warnings are to be ignored.

Create RIB Objects

1. Change directories to INSTALL_DIR/rib_objects/db_objects.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rms121rib.sql
   ```
3. Check the log file rms121rib.log for any errors noting that ORA-04043 errors are to be ignored.

Loading RIB data

1. Change directories to INSTALL_DIR/rib_objects/ddl.
2. Run the following command at the UNIX prompt:
   ```
   sqlldr RMS121DEV/SCHEMA_PASSWORD control=rib_doctypes_rms.ctl
   ```
3. Check the log file rib_doctypes_rms.log for any errors.

Create Pricing Objects

Create Pricing Types

1. Change directories to INSTALL_DIR/pricing/types.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rpm121type.sql
   ```
3. Check the log file rpm121type.log for any errors noting that ORA-04043 errors and warnings are to be ignored.

Create Pricing DDL

1. Change directories to INSTALL_DIR/pricing/ddl.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rpm121.sql
   ```
3. Check the log file rpm121.log for any errors.

Create Pricing Objects

1. Change directories to INSTALL_DIR/pricing/db_objects.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rpm121dbo.sql
   ```
3. Check the log file rpm121dbo.log for any errors noting that ORA-04043 errors are to be ignored.

Create RMS Objects

Create RMS types

1. Change directories to INSTALL_DIR/types.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rms121type.sql
   ```
3. Check the log file rms121type.log for any errors

Create RMS Tables

Some views and triggers that depend on packages and procedures are created later in the install. Some views give compilation warnings. These are OK. The warnings are caused by the views dependency on a package that are installed later. The warnings are cleared when objects are revalidated later in the install.

1. Change directories to INSTALL_DIR/ddl.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rms121.sql
   ```
3. Check the log file rms121.log for any errors.

Create RMS Database Objects

1. Change directories to INSTALL_DIR/db_objects.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rms121dbo.sql
   ```
3. Check the log file rms121dbo.log for any errors.

Create Remaining RMS Views and Triggers

1. Change directories to INSTALL_DIR/ddl.
2. Log into sqlplus as RMS121DEV and run the following command:
   ```
   SQL> @rms121b.sql
   ```
3. Check the log file rms121b.log for any errors.

Validate all Invalid Objects

> **Note:** Deadlocked objects may appear when running this script. This is expected. Run the script until no invalid objects remain.

1. Change directories to INSTALL_DIR/utility.
2. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @inv_obj_comp.sql`
3. This script may need to be run more than once.

Insert Data for Pricing

1. Change directories to INSTALL_DIR/pricing/data.
2. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @rpm121ctl.sql`
3. Check the log file rpm121ctl.log for any errors.
4. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @rsm121rpm.sql`
5. Check the log file rsm121rpm.log for any errors.

Insert Data for RMS

> **Note:** This script prompts for a role. The default is developer.

1. Change directories to INSTALL_DIR/data.
2. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @rms121ctl.sql`
3. Check the log file rms121ctl.log for any errors.

Insert Demo Data for RMS

> **Note:** Running this script provides some demo data such as stores and items.

1. Change directories to INSTALL_DIR/data
2. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @rmsdemodata.sql`
   The following items will be prompted for:
   - How many characters country codes should be? 2 or 3?
   - Is Multi Channel on?
   - Is VAT on?
   - Is Class Level Vat on?
   - Is Bracket Costing on?
   - Name of the RMS schema owner
   - Primary currency (ex USD for US Dollar or EUR for the Euro)
   - Is ELC inclusive on?
   - How many characters country codes should be? 2 or 3:

- Number of demo items to create
- Transaction level for these items:
  - Line
  - Line Extension
  - Variant

3. Check the log file rmsdemodata.log for any errors.

Insert Additional Data for RMS

> **Note:** If demo data was not run please insert values into the SYSTEM_OPTIONS table.

1. Change directories to INSTALL_DIR/data.
2. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @rms121ctlb.sql`
3. Check the log file rms121ctlb.log for any errors.

Insert RTM Data

> **Note:** This is for Oracle Trade Management Customers. For more information on these data scripts please refer to Appendix I.

1. Change directories to INSTALL_DIR/data/rtm.
2. Log into sqlplus as RMS121DEV and run the following command:
   `SQL> @rms121rtm.sql`
3. Check the log file rms121rtm.log for any errors.

Compile RMS Batch Libraries and Programs

If the platform for this installation is AIX, refer to Appendix E, "AIX Shared Library Bug Fix". This is a mandatory fix on AIX platforms in order for RMS batch programs to function properly.

> **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/rms.
2. Set the following variables:

> **Note**: INSTALL_DIR is the location where RMS 12 will be installed.
>
> Make sure the path for make, makedepend, and the compiler are in $PATH environment variable.

- 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

Configure Make File

1. As the oretail user, change directories to INSTALL_DIR/rms/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_aix_64bit.mk platform.mk

3. Run the oramake script from INSTALL_DIR/rms/oracle/lib/src directory. This will use 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. Change directories to INSTALL_DIR/db_objects.
2. Log into SQL*Plus as RMS121DEV and run the following scripts:
   - SQL> @createordlib.sql
   - SQL> @dealinclib.sql
   - SQL> @dealordlib.sql
   - SQL> @scllib.sql
3. Exit SQL*Plus.

Re-Validate RMS Database Objects

1. As the oretail user, change directories to INSTALL_DIR/utility
2. Log into SQL*Plus as RMS121DEV 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/rms/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/rms/oracle/lib/bin

Compile Batch Source Code

1. As the oretail user, change directories to INSTALL_DIR/rms/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/rms/oracle/proc/bin

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, 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 will have access to.
3. Change directories to INSTALL_DIR/data and run the following scripts as the new user to give new users security privileges.

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

3

Application Server Installation Tasks

It is assumed that Oracle Application Server 10g version 10.1.2.0.2 (OAS) has already been installed. If not, refer to the Check Application Server Requirements section in the Preinstallation Tasks chapter before proceeding. Additionally, *INSTALL_DIR* in this section refers to the directory created in Create Staging Directory for RMS Application Files in the Preinstallation Tasks chapter.

In order to use Forms Builder 10g for manual compilation of RMS 12 forms modules, Oracle Developer Suite 10g Release 2 (10.1.2.0.2) must be used. It should be noted that Oracle has not released a version of 10gDS for AIX 5.3 platform. 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 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 (10.1.2.0.2) has been installed

1. The T2kMotif.rgb file that is sent out with Oracle Application Server 10g (10.1.2.0.2) 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. 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 (10.1.2.0.2) has been installed

   All OS Platforms

   - PATH=$ORACLE_HOME/bin:$ORACLE_HOME/opmn/bin:$ORACLE_HOME/dcm/bin:INSTALL_DIR/forms10gr2_scripts:$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=$CLASSPATH
- FORMS_PATH=INSTALL_DIR/toolset/bin:INSTALL_DIR/rms/forms/bin:$ORACLE_HOME/forms
- REPORTS_PATH=INSTALL_DIR/rms/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 12 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

RMS Toolset Installation

1. Copy all libraries (.pll files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory.
2. Change directories to INSTALL_DIR/toolset/bin.
3. Run f10gr2plsqlconv_pll_stand45 to automatically attach the Forms 10g library rp2rro.pll to stand45.pll. This library must be attached to stand45.pll in order to run RMS reports.
4. Remove the newly created stand45.pld should it be created from running f10gr2plsqlconv_pll_stand45.
5. 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

6. Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully).
7. Remove all newly created .plx files.
8. Copy all forms (*.fmb files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory.
9. Run fmb2fmx10gr2_fm (in INSTALL_DIR/toolset/bin) to compile the Toolset reference forms.
10. Remove all newly created fm_*.fmx files (reference forms should not have executable files).
11. Run fmb2fmx10gr2 (in INSTALL_DIR/toolset/bin) to generate Toolset runtime forms – .fmx's.
12. 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/toolset/bin directory.

13. Remove all non-reference form forms from INSTALL_DIR/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
```

14. Copy all menus (*.mmb files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory.
15. Run mmb2mmx10gr2 (in INSTALL_DIR/toolset/bin) to generate Toolset runtime menus – .mmx's.
16. 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.

17. Remove all .mmb files from INSTALL_DIR/toolset/bin.

RMS Forms Installation

1. Copy all libraries (.pll files) in the INSTALL_DIR/rms/forms/src directory to the directories to the INSTALL_DIR/rms/forms/bin directory.
2. Change directories to INSTALL_DIR/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/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory.
6. Run fmb2fmx10gr2_fm (in INSTALL_DIR/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/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/rms/forms/bin directory.

10. Remove all non-reference form forms from INSTALL_DIR/rms/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
```

11. Copy all menus (*.mmb files) in the INSTALL_DIR/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory.
12. Run mmb2mmx10gr2 (in INSTALL_DIR/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/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.

RMS Reports Installation

1. Copy the reports library (rep25lib.pll) in the INSTALL_DIR/rms/reports/src directory to the INSTALL_DIR/rms/reports/bin directory.
2. Change directories to INSTALL_DIR/rms/reports/bin.
3. Run pll2plx10gr2_reports to compile rep25lib.pll.
4. Remove the newly created rep25lib.plx file.
5. Copy all reports (*.rdf files) in the INSTALL_DIR/rms/reports/src directory to the INSTALL_DIR/rms/reports/bin directory
6. Run rdf2rep10gr2 (in INSTALL_DIR/rms/reports/bin) to generate Reports runtime reports – .rep's.

Note: The following error messages may appear when running rdf2rep10gr2; these errors can be ignored if report (.rep) generation was successful:

REP-0759: One or more PL/SQL libraries have been modified since the reports was saved. The PL/SQL will be recompiled.

REP-0202: Attempt to free a null pointer

REP-0759 is generated by the r25conv program. The error appears any time a report is converted.

REP-0202 is due to an Oracle bug with rwconverter and can be ignored

7. Check to make sure that each .rdf file has a corresponding .rep file.
8. Remove all .rdf files from INSTALL_DIR/rms/reports/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 (10.1.2.2) components must be started in order to run Oracle Forms applications.

Note: ORACLE_HOME refers to the location where Oracle Application Server 10g (10.1.2.2) Forms and Reports Services is installed.

Note: Prior to modifying Oracle Application Server 10g (10.1.2.2) 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. Additionally, 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.

Example:
FORMS_PATH=/u00/rms/toolset/bin:/u00/rms/forms/bin:/u00/oracle/AS10GR2/forms

NLS_DATE_FORMAT=DD-MON-RR

NLS_LANG=AMERICAN_AMERICA.UTF8

FORMS_REJECT_GO_DISABLED_ITEM=FALSE

A Reports Server needs to be running in order to access RMS 12 reports through the RMS 12 web environment; either the default reports server can be used, or a new reports server can be used.

A default reports server was created and started during the Oracle Application Server 10g version 10.1.2.0.2 installation; at that time a 10g reports server entry was automatically made in ORACLE_HOME/network/admin/tnsnames.ora; the name of the default reports server is REP_<SERVER_NAME>.

A new reports server can be created by running the script ORACLE_HOME/bin/rwserver.sh; all variables required for compiling 10g reports must be set, and there must be a 10g reports server entry in ORACLE_HOME/network/admin/tnsnames.ora prior to running rwserver.sh. In order to make a new 10g reports server entry in tnsnames.ora, make a copy of the default 10g reports server entry, changing the name and port (un-used port on the server). rwserver.sh must be run specifying the 10g reports server entry in the tnsnames.ora file.

Example (tnsnames.ora): REP_RMS12 =
(ADDRESS = (PROTOCOL = tcp)(HOST = server)(PORT = 1951))

Example: rwserver.sh server=REP_RMS12

Verify the following resulted from running rwserver.sh:

10g reports server process started for the reports server specified configuration file was created in ORACLE_HOME/reports/conf for the reports server specified (named REP_RMS12.conf)

Note: Contact Oracle Support for problems with starting a 10g reports server. For disabling Single Sign On security with 10g reports, Oracle recommends removing the security tag from the REP_<SERVER_NAME>.conf file. Doing so can also solve problems with starting a 10g reports server.

3. Modify the file ORACLE_HOME/bin/reports.sh by appending the location of the RMS reports modules to the REPORTS_PATH variable setting.

Example:
REPORTS_PATH=/u00/rms/reports/bin:$ORACLE_HOME/reports/templates:$ORACLE_HOME/reports/samples/demo:$ORACLE_HOME/reports/integ:$ORACLE_HOME/reports/printers

4. 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 12 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.
5. Log into sqlplus as the RMS 12 schema owner (RMS121DEV) 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 exit

WEBREPORTS_SERVER='reports/rwservlet' where lang=1';

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

6. 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, separateFrame applet parameters; and starting form for the RMS application.

Example: [rms]

envfile=rms.env

width=850

height=585

separateFrame=true

form=rtkstrt.fmx

7. Additional modifications are needed to ensure that RMS utilizes the Sun JRE plug-in installed on the client. Comment out the following lines in formsweb.cfg at the beginning of this file:
   - baseHTMLjinitiator=basejini.htm
   - baseHTMLjpi=basejpi.htm

Example: ## baseHTMLjinitiator=basejini.htm
baseHTMLjpi=basejpi.htm

Add the following lines after the "Single Sign-On OID configuration parameter" section of formsweb.cfg . This directs clients to use the latest version of the Sun Java Plug-in installed on their machine when accessing RMS. No update is needed if you are using a different minor version of the Java plug-in.

##
#########

```
## added for Java 1.4.1+

## Use this classid to allow users to use any 1.4.X plugin

jinit_classid=clsid:8AD9C840-044E-11D1-B3E9-
00805F499D93

jinit_mimetype=application/x-java-applet;jpi-
version=1.4.1_03

legacy_lifecycle=true

                    ## end Java plug-in additions

    ################################################
#############
```

8. 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/

9. If NLS_LANG is NOT set in the ORACLE_HOME/forms/server/rms.env then copy the RMS keyboard-mapping file INSTALL_DIR/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/sample_files/fmrweb_utf8.res to ORACLE_HOME/forms/admin/resource/US.

10. Copy the sample file INSTALL_DIR/sample_files/rms12unix.conf to ORACLE_HOME/Apache/Apache/conf. rms12unix.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.

11. In rms12unix.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/.

12. Add the contents of rms12unix.conf to the end of httpd.conf, or add an include directive in httpd.conf to rms12unix.conf.

13. 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.

14. 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 6 above).

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

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

15. 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: RMS121DEV
    Password: retek
    Connect String: prod_db1

A

Appendix: Oracle 10g Database Creation Scripts

```
###############################################################################
# Oracle 10.2.0.x Parameter file
#
# NOTES: Before using this script:
#        1.  Change <datafile_path>, <admin_path>, <utl_file_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
# +------+ +---------+ +-----------------+ +-------------+ +-------------+
# 02/20/06 Oracle       NA                  NA              creation
#
###############################################################################
# ------------------------------------------------------------------------------
# The following SGA parameters are CRITICAL to the performance of the
# database. The following settings are based on 1GB of allotted memory.
# The SGA is composed of:
#    db_cache_size, log_buffer,java_pool_size, large_pool_size, shared_pool_size
# ------------------------------------------------------------------------------
db_cache_size                  = 256M
java_pool_size                 = 150M        # 150M for initial db creation
log_buffer                     = 10485760
shared_pool_size               = 350M        # 350M for initial db creation
shared_pool_reserved_size      = 35M         # 10% of shared_pool_size
# ------------------------------------------------------------------------------
# The following parameters do not affect SGA size;
# ------------------------------------------------------------------------------
audit_file_dest               = <admin_path>/adump
background_dump_dest          = <admin_path>/bdump
compatible                    = 10.2.0
control_files                 = (<datafile_path>/control01.ctl
                                ,<datafile_path>/control02.ctl)
core_dump_dest                = <admin_path>/cdump
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
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 prop
er 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     = 10.2.0.1
optimizer_mode                = CHOOSE    # Oracle Retail required
pga_aggregate_target          = 100M
plsql_optimize_level          = 2         # 10g change; use this setting to
optimize plsql performance
plsql_debug                   = false     # 10g change; use this setting to
optimize plsql peformance
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; 10g uses to
cache sql cursors in pl/sql
undo_management               = AUTO
undo_retention                = 1800      # Currently set for 30 minutes; set to
avg length of transactions in sec
undo_tablespace               = undo_ts
user_dump_dest                = <admin_path>/udump
utl_file_dir                  = <utl_file_path>
workarea_size_policy          = auto      # Should be set to auto when
pga_aggregate_target is set

# ***  Set these parameters for Oracle Retail Data Warehouse (RDW) database ***
#nls_date_format                        = DD-MON-RRRR  # Required by MicroStrategy
#query_rewrite_integrity                = TRUSTED
#star_transformation_enabled            = TRUE
#utl_file_dir                     = <Windows_utl_file_path>, <UNIX_util_file_path>

# ***  Archive Logging, set if needed  ***
#log_archive_dest_1           = 'location=<admin_path>/arch/'
#log_archive_format           = SIDarch_%r_%s_%t.log
#log_archive_max_processes    = 1         # Default:1
#log_archive_min_succeed_dest = 1         # Default:1
#log_buffer                   = 262144    # Set to (512K or 128K)*CPUs
#log_checkpoint_interval      = 51200     # Default:0 - unlimited
#log_checkpoint_timeout       = 7200      # Default:1800 seconds

-------------------------------------------------------------------------------
--- Script:      crdb1.sql
--- Execute as:  sysdba
--- Note:        Before running this script:
---                 Modify <datafile_path> values.
---                 Modify SID values.
---                 Adjust sizes for redo logs, datafiles and tempfile
-------------------------------------------------------------------------------
spool crdb1.log
STARTUP NOMOUNT pfile=${ORACLE_HOME}/dbs/initSID.ora
CREATE DATABASE "SID"
        MAXDATAFILES 1000
        CHARACTER SET UTF8
    DATAFILE
        '<datafile_path>/system01.dbf' SIZE 500M AUTOEXTEND ON NEXT 100M MAXSIZE
2000M
    LOGFILE
      GROUP 1 ('<datafile_path>/redo1a.log') SIZE 1000M,
      GROUP 2 ('<datafile_path>/redo2a.log') SIZE 1000M,
      GROUP 3 ('<datafile_path>/redo3a.log') SIZE 1000M
DEFAULT TEMPORARY TABLESPACE temp TEMPFILE '<datafile_path>/temp01.dbf' SIZE 5000M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M
UNDO TABLESPACE undo_ts DATAFILE '<datafile_path>/undo_ts01.dbf' SIZE 5000M
```

```
SYSAUX DATAFILE '<datafile_path>/sysaux01.dbf' SIZE 500M AUTOEXTEND ON NEXT 100M
MAXSIZE 2000M
;
exit
spool off
------------------------------------------------------------------------------
--- Script:       crdb2.sql
--- Execute as:   sysdba in 10.1.0.2 databases or higher
--- Note:         This script installs the data dictionary views in addition to
---               granting necessary privileges to public.
------------------------------------------------------------------------------
spool crdb2.log
REM # install data dictionary views:
PROMPT Running catalog.sql
@$ORACLE_HOME/rdbms/admin/catalog.sql;
PROMPT Running catblock.sql
@$ORACLE_HOME/rdbms/admin/catblock.sql;
PROMPT Running catproc.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql;
PROMPT Running catoctk.sql
@$ORACLE_HOME/rdbms/admin/catoctk.sql;
PROMPT Running catrep.sql
@$ORACLE_HOME/rdbms/admin/catrep.sql;
PROMPT Running owminst.plb
@$ORACLE_HOME/rdbms/admin/owminst.plb;

REM * These privs needed for users to run proper grant code when creating users.
grant select on dba_jobs to public with grant option;
grant select on dba_roles to public with grant option;
grant select on dba_role_privs to public with grant option;
grant execute on dbms_rls to public with grant option;
grant execute on dbms_alert to public;
grant select_catalog_role to public;
grant execute_catalog_role to public;
grant execute on dbms_lock to public;
grant execute on dbms_rls to public;
grant select any dictionary to public;


REM * query rewrite privilege needed to create function-based indexes
grant query rewrite to public;

REM * dbms_system is needed for tracing
grant execute on sys.dbms_system to public;

PROMPT Creating PLAN table owned by SYSTEM
@$ORACLE_HOME/rdbms/admin/utlxplan.sql

PROMPT Creating public synonyn for the plan table
create public synonym PLAN_TABLE for SYSTEM.PLAN_TABLE;

connect SYSTEM/manager
@$ORACLE_HOME/sqlplus/admin/pupbld.sql;
@$ORACLE_HOME/sqlplus/admin/help/hlpbld.sql helpus.sql;

spool off
exit
```

```
----------------------------------------------------------------
--- Script:        crdb3.sql
--- Execute as:    sysdba in 10.1.0.2 databases or higher
--- Note:          This script installs java and xml components;
---                Do not change the order of the statements below due to
--                 dependencies
----------------------------------------------------------------
spool JServer.log
@$ORACLE_HOME/javavm/install/initjvm.sql;
@$ORACLE_HOME/xdk/admin/initxml.sql;
@$ORACLE_HOME/xdk/admin/xmlja.sql;
@$ORACLE_HOME/rdbms/admin/catjava.sql;
@$ORACLE_HOME/rdbms/admin/catexf.sql;
spool off

spool context.log
@$ORACLE_HOME/ctx/admin/catctx change_on_install SYSAUX TEMP NOLOCK;
connect CTXSYS/change_on_install
@$ORACLE_HOME/ctx/admin/defaults/dr0defin.sql AMERICAN;
spool off

spool xdb_protocol.log
connect / as sysdba
@$ORACLE_HOME/rdbms/admin/catqm.sql change_on_install SYSAUX TEMP;
spool off
```

@$ORACLE_HOME/rdbms/admin/utlrp.sql

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 MANUAL
;
CREATE TABLESPACE RETEK_DATA DATAFILE
    '<datafile_path>/retek_data01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 500M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT MANUAL
;
CREATE TABLESPACE LOB_DATA DATAFILE
    '<datafile_path>/lob_data01.dbf'  SIZE 50M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT MANUAL
;
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,
      drop any synonym,
      execute any procedure,
      execute any type,
      select any sequence,
      select any table,
      select any dictionary,
      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: AIX Shared Library Bug Fix

The env_rdbms.mk file for Oracle 10g 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)

-----------------------------------------------
```

F

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 *RETL 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 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 variables that need to change depending on local settings:

```
export DBNAME=int9i
export RMS_OWNER=RMS121DEV
export BA_OWNER=rmsint1012
```

In addition, you 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 causes the password "bogus" to be used to log into the database:

```
export PASSWORD=pass1
```

G

Appendix: Oracle Trade Management 12 System Expectations

Install 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 is 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 will be 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 will be 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 an Item 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 will be 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 will be 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. 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 will have 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 will be 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. Therefore, 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. Therefore, 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 are 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 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 is 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 four 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 sets 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. In addition, 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 has 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.

H

Appendix: RMS-RIB Custom Post – Processing

The following are instructions for installing RMS – RIB Custom Post Processing. This should be completed after the RIB has been installed.

- Un-tar the tarfile into the RIB INSTALL subdirectory.
- This will create a RIBCustPostProcXXXX subdirectory.
- Copy the jar file desired from this subdirectory (there are currently two custom postprocessing jar files) into both of the subdirectories.
  - $EHOME/client/classes
  - $EHOME/server/registry/repository/<RIB SCHEMA>/runtime/classes
- Rename the jar file to custom-postprocess-impl.jar