Retail passion. Innovative solutions.

Retek® Merchandising System™ 11.0

Installation Guide

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA

888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

Sales Enquiries:
+44 (0)20 7563 46 46

Fax:
+44 (0)20 7563 46 10

The software described in this documentation is furnished under a license agreement, is the confidential information of Retek Inc., and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403, and the copyright notice may not be removed without the consent of Retek Inc.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

Retek® Merchandising System™ is a trademark of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2004 Retek Inc. All rights reserved.

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method	Contact Information
E-mail	support@retek.com
Internet (ROCS)	rocs.retek.com Retek's secure client Web site to update and view issues
Phone	+1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835

Mail

Retek Customer Support
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403

When contacting Customer Support, please provide:

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

Contents

Chapter 1 – Hardware and Software Requirements

Prior to installing Retek Merchandising products, review the requirements listed below. Verify that these requirements are met and that the hardware will adequately run the software to be installed, as well as process the anticipated volume of data.

Supported Media	Retek Merchandising is available on the Retek Fulfillment Center Web Site, https://fulfillment.retek.com/.
Database Server	Database software requirements.
Application Server	Operating system software and development tools and a list of hardware choices.
Web Browser	Supported OS/Browser/Java plug-in requirements

The following table lists acceptable operating system versions and web browser versions.

	Database Server	Application Server	Client
Vendor	Oracle RDBMS 9i - Enterprise Edition	Oracle 10G Application Server Forms and Reports Services	Web Browser (IE) Sun JRE plugin 1.4.1+
Retek (RMS 11.x)	Batch Libraries DDL (Views, Triggers, Tables etc.) Database Objects (Procedures, Packages) Control Scripts Data Scripts	Forms Reports Toolset Help Files	

Retek Merchandising System

Database Server

General Requirements for a database server running RMS include:

- Unix based OS(AIX, Solaris, or HP-UX PA RISC) certified with Oracle 9i
- ANSI compliant C compiler
- Perl compiler 5.0 or later
- Oracle RDBMS 9i Enterprise Edition (minimum 9.2.0.5 patchset required)
  - Sun platform requires patch for bug 3566662
  - AIX 5.2 platform requires patch for bug 2820694
- Oracle Partitioning
- Oracle Pro*C Precompiler (per database version)
- Oracle Net Services
- x-Windows interface

For development:

- OCI
- Oracle XML Developers Kit
- Oracle XML SQL Utility

Hardware/OS options as used for development (see Oracle's website for certified platforms):

- Sun Solaris 9
- IBM/AIX 5.2
- Hewlett Packard/HP UX 11.11

Application Server

General requirements for an application server running RMS include:

- Unix based OS(AIX, Solaris, or HP-UX PA RISC) certified with Oracle 10G Application Server 9.0.4.0
- x-Windows interface

Sizing factors and other suggestions to factor into hardware configuration of application server include:

- CD-ROM drive
- 1 GB network adapter
- ~3 GB Free disk space for 9iAS
- ~1 GB Free disk space for RMS forms, reports, gif files and help files.

Hardware/OS options as used for development

- Sun Solaris 9
- IBM/AIX AIX 5.2
- HP-UX 11.11

Web Browser and Client requirements

General requirements for client running RMS include:

JRE Plugin:

- Sun JRE Plug-in 1.4.1+

Client PCs:

- Pentium Processor
- Windows 2000 or XP
- 1024x768 resolution

Sizing factors and other suggestions to factor into selection of a PC include:

- PC Configuration (minimum 256 MB RAM, 450 MHz)

Browser options to factor into selection include:

- Internet Explorer 5.5, 6.0 and higher

Chapter 2 – Database Installation Instructions

Database Server Installation Instructions

Create a UNIX user account to install the software

1 Create a UNIX group named “dev”.

2 Create UNIX user named “retek” 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 UNIX server as retek.

2 Create a staging directory for the RMS database installation software. There should be a minimum of 100 MB disk space available.

3 Copy the rms11dbserver.zip file from the CD/dbserverunix directory to the staging directory. This will be referred to as INSTALL_DIR for the remainder of this chapter.

4 Change directories to INSTALL_DIR and extract the rms11dbserver.zip file.

Establish a Partitioning Strategy

For optimal performance of the RMS application, partitioning is mandatory for specific tables. Requirements for mandatory and optional partitioning are defined in the Microsoft Excel spreadsheet 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 INSTALL_DIR/ddl/part/RMS_partition_definition.xls to determine an appropriate partitioning strategy. 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/rms11_part.tab or by implementing the process defined in Appendix A. This file will be used later in the installation process.

Note: Refer to Oracle9i Database Concepts Release 2 (9.2) Chapter 11 “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 following fields at the top of the RMS worksheet. Altering these values will update the “Number of Partitions” column for HASH partitioned/sub-partitioned tables. The value in this column indicates the number of hash partitions/sub-partitions to create.

Locations: The total number of active stores and warehouses.

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.

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.

Create the Oracle 9i Database

Note: It is assumed that Oracle 9i release 2 with appropriate patches has already been installed. If not, refer to "Database Server" requirements and install before proceeding.

Note: If a database has already been created, it is necessary to review the contents of each script noted below to determine if all database components have been installed and configured properly. Refer to Appendix B.

Create the database as follows:

1 Login to UNIX as the oracle user.

2 Create the directory structure for the database (datafile directories, adump, bdump, cdump, arch, 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 $ORACLE_HOME/dbs directory and rename it to init${ORACLE_SID}.ora. Modify the parameters according to guidelines specified in this file.

5 Modify the INSTALL_DIR/create_db/crdb1.sql file. Refer to comments in this file regarding modifications that need to be made.

6 Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/crdb1.sql. Review crdb1.log for errors and correct as needed.

7 Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/crdb2.sql. Review crdb2.log for errors and correct as needed.

8 Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/crdb3.sql. Review crdb3.log for errors and correct as needed.

9 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 C.

Create RMS Tablespaces

1 Modify INSTALL_DIR/create_db/create_rms_tablespaces.sql. Refer to Appendix D.

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 Retek Services.

2 Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/create_db/create_rms_tablespaces.sql. Review create_rms_tablespaces.log for errors and correct as needed.

Create RMS Schema Owner

Create an Oracle schema that will own the RMS application. Refer to Appendix E.

1 Change directories to INSTALL_DIR/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. The following will be prompted for:

Schema Owner – the Oracle user that will own all RMS objects. Referred to in this install guide as RMS11DEV

Password – the password for RMS11DEV

Temp Tablespace – the temporary tablespace for RMS11DEV

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

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 for RMS

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.

Note: The following directories are included but not used

CastorPayloadTyped - Contains typed, serialized java beans representing message families. A configuration file (payload.properties) maps each bean to a specific message family/message type.

CastorPayloadUntyped - Contains untyped, serialized java beans representing message families. Each class member is represented by a String (as opposed to the data type the member represents). A configuration file (payload.properties) maps each bean to a specific message family/message type.

Retek_Pub_Trans - Contains a class that maps an oracle object to an XML formatted string for every family represented in a database by an oracle object. Each translator handles all message types within a single family

Create RIB tables and types

1 Change directories to INSTALL_DIR/rib_objects110/xml

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11xml.sql
```

3 Check the log file rms11xml.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_objects110/Oracle_Objects.

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11rib.sql
```

3 Check the log file rms11rib.log for any errors noting that ORA-04043 errors are to be ignored.

Loading RIB data

1 Change directories to INSTALL_DIR/rib_objects110/xml.

2 Run the following command at the UNIX prompt:

```
sqlldr RMS11DEV/SCHEMA_PASSWORD control=rib_doctypes_rms.ctl
```

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

Create RPM Objects for RMS

Refer to the RPM Initial Price Setting Install Guide for this portion. Complete the database section before continuing on.

Create RMS Objects

Note: When running the scripts in this section the following errors may be encountered "Warning: Type created with compilation errors". These errors can be ignored. 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 RMS tables

Note: If Sales Audit is not going to be installed run the script rms11wosa.sql instead of rms11.sql

Note: Some views and triggers that depend on packages and procedures will be created later in the install

Note: Three views will give compilation warnings. These are OK. The warnings are caused by the views dependency on a package that will be installed later. The warnings will be cleared when objects are revalidated later in the install.

1 Change directories to INSTALL_DIR/ddl

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11.sql
```

3 Check the log file rms11.log for any errors.

Create toolset database objects

1 Change directories to INSTALL_DIR/toolset_dbo

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11toolset.sql
```

3 Check the log file rms11toolset.log for any errors.

Create RMS database objects

1 Change directories to INSTALL_DIR/db_objects

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11dbo.sql
```

3 Check the log file rms11dbo.log for any errors.

Create remaining views and triggers

1 Change directories to INSTALL_DIR/ddl

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11b.sql
```

3 Check the log file rms11b.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 more invalid objects remain.

1 Change directories to INSTALL_DIR/utility

2 Log into sqlplus as RMS11DEV and run the following command:

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

3 This script may need to be run more than once.

Insert data for RMS

Note: This script will prompt for a role. The default is developer.

1 Change directories to INSTALL_DIR/sqlplus

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11ctl.sql
```

3 Check the log file rms11ctl.log for any errors.

Insert demo data for RMS

Note: Running this script is optional. It provides some demo data such as stores and items.

1 Change directories to INSTALL_DIR/sqlplus

2 Log into sqlplus as RMS11DEV and run the following command:

- SQL> @rms11demodata.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)
- 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 rms11demodata.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/sqlplus

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11ctlb.sql
```

3 Check the log file rms11ctlb.log for any errors.

Insert RTM data

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

1 Change directories to INSTALL_DIR/sqlplus/rtm

2 Log into sqlplus as RMS11DEV and run the following command:

```
SQL> @rms11rtm.sql
```

3 Check the log file rms11rtm.log for any errors

Compile RMS batch libraries and programs

Note: If compiling on AIX, refer to Appendix G.

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

Setting Environment Variables

1 Change directories to INSTALL_DIR/rms

2 As the retek user, set the following variables:

Note: INSTALL_DIR is the location where RMS 11 will be installed.

Note: 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 only:

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

HP only:

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

Solaris only:

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

3 Change directories to INSTALL_DIR/rms/oracle/lib/src

4 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

5 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.

6 Change directories to INSTALL_DIR/db_objects

7 Four sql files must be run after MMHOME has been set.

Log into sqlplus as RMS11DEV and run the following scripts:

- SQL> @createordlib.sql
- SQL> @dealinclib.sql
- SQL> @dealordlib.sql
- SQL> @scllib.sql

8 Exit from sqlplus

9 Change directories to INSTALL_DIR/utility

10 Log into sqlplus as RMS11DEV and run the following command:

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

This script may need to be run more than once.

11 Change directories to INSTALL_DIR/rms/oracle/lib/src and run the following commands

12 To make library dependencies

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

13 Check the libdpnd.log file for errors

14 To make batch libraries

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

15 Check the libretek.log file for errors

16 To install batch libraries

```
make –f retek.mk install
```

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

18 Change directories to INSTALL_DIR/rms/oracle/proc/src and run the following commands

19 To make dependencies

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

20 Check the srcdpnd.log file for errors

21 To make batch programs

Because of an additional make command the following command must be run first

```
make –f rms.mk PRODUCT_PROCFLAGS=dynamic=ansi ditinsrt
```

To make the rest of the batch programs run the following command

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

22 Check the srcall.log file for errors

23 To install batch programs

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

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

Setting Up additional Users

If additional ORACLE users are to be set up at this time, permissions will need to be granted to them so they can run RMS. During integration, there should be multiple roles set up and assigned appropriately to users, based on user responsibilities.

1 Until that point, however, use the following grant command as a sample for what privileges should be granted to users:

SQL> 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>;

SQL> grant developer to <userid>;

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 Finally, change directories to INSTALL_DIR/sqlplus and run the following scripts as the new user to give new users security priviledges.

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

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

Chapter 3 – Application Server Installation Instructions

Note: INSTALL_DIR is the directory where the RMS 11.x files will be extracted to. 9iAS10G_ORACLE_HOME is the location where Oracle 9iAS 10g Forms and Reports Services will be installed.

UNIX (Sun Solaris/HP-UX/AIX)

Pre-installation Checklist

- Verify the system hosts file contains the fully qualified domain name of the server
- Oracle Bug 3713307 (exclusion of rootpre.sh in Oracle 9iAS Forms and Reports Services v9.0.4 for AIX 5L) requires that rootpre.sh from Oracle 9iAS Enterprise Edition 9.0.4 AIX 5L Disk 1 be run in order to set kernel tunable parameters and check asynchronous I/O
- For installations on the AIX 5L Operating System, download Java(TM) Cryptography Extension 1.2.2 (JCE) software from http://java.sun.com/products/jce/index-122.html. This software is required for Oracle 9iAS 10g Forms and Reports on AIX 5L. Place the software (jce-1_2_2.zip) in a staging directory on the application server, as it will need to be accessed by the Oracle installer.

Note: The following variables should NOT be set prior to installing 9iAS 10g: ORACLE_HOME, LD_LIBRARY_PATH, CLASSPATH.

Note: The variables TMP and TEMP should be set to a location with at least 1 GB of free space. The DISPLAY variable must be set to the IP Address plus ":0.0" (ie: 10.1.1.1:0.0) of the client machine that is being used to perform the installation.

Install Oracle 9iAS 10g (9.0.4.0) Forms and Reports Services

Note: Oracle installation tools vary by platform. Essential 9iAS 10g Forms and Reports Services information is provided below, but additional questions/options may be presented during the installation. In these cases use Oracle's default setting or consult Oracle support. The Oracle 9iAS 10g Forms and Reports Services installation requires approximately 2 GB of free space.

1 Create a UNIX user to be used as the Oracle 9iAS 10g administrator account. The 9iAS 10g administrator user must belong to the dba group. Log into the application server as the 9iAS 10g administrator user (example: oracle:dba).

2 Insert the Oracle9i Application Server Forms and Reports Services 10g CD 1 into the CD-ROM.

3 Start the Oracle installer from a location other than in the /cdrom drive. Do not run the runInstaller script in the background (i.e. /runInstaller) as this may cause the script to fail.

Example: /u00/oracle> ./cdrom/runInstaller

4 If prompted, run rootpre.sh from Oracle 9iAS Enterprise Edition 9.0.4 Disk 1 as the root user.

5 If prompted, enter the staging directory on the application server where the jce-1_2_2.zip file is located.

Example: /u00/oracle/staging

6 If all system requirements are met, the Oracle GUI Installer will be launched.

7 On the Welcome page, click Next.

8 On the File Locations page, verify that the Source and Destination fields are correct, and enter a unique Oracle Home name, and then click Next.

Note: Oracle 9iAS 10g cannot share an ORACLE_HOME with another Oracle product.

Example: Name: 9iAS10G
Path: /u00/oracle/9iAS10G

9 In the Language Selection window, select desired languages that the Oracle 9iAS 10g Forms and Reports instance may run in, and click OK, and then click Next.

10 On the Create Instance Name and ias_admin Password page, enter an Instance Name and a Password for the ias_admin user, and then click Next.

Example: Instance Name: 9iAS10G
ias_admin Password: oracle10g
Confirm Password: oracle10g

11 If the Choose JDK Home Directory page appears, enter a JDK 1.4.1 home and then click Next.

12 On the Provide Outgoing Mail Server Information page, leave all fields blank and then click Next.

13 On the Summary page, check the product list and then click Install.

14 When prompted, run 9iAS10G_ORACLE_HOME/root.sh as the root user, and then click OK after this script has been run.

Note: The default values (for Oracle script/file locations) presented while running root.sh should be accepted unless there is a specific reason to override the defaults. Do NOT override any default values unless fully aware of the ramifications of doing so.

15 The Configuration Tools page appears next, with numerous products that attempt to start up automatically. The Oracle Net Configuration Assistant may appear as well. Do not click Stop on this window, but rather let the Configuration Assistant complete.

Note: An error window may appear stating that one or more of the tools have failed. If this happens, view the details of the error in the Details window and attempt to troubleshoot, or contact Oracle Support. Do not leave tools in failure status unless it's certain the failed tools will not be needed in production.

16 The End of Installation window will appear when all components on the Configuration Tools page have started up successfully. Record the information from this window for future reference. Specifically, the URL to access the Oracle Enterprise Manager (OEM) Application Server Control will be needed when configuring Oracle 9iAS 10g for RMS.

17 Click Exit to exit the Oracle installer.

18 To load OEM and view the Oracle 9iAS 10g System Components, load the OEM URL from step 16 above. If the OEM URL was not recorded, the default OEM URL should be **Error! Hyperlink reference not valid.**. Alternatively, the OEM port can be obtained from the file 9iAS10G_ORACLE_HOME/install/portlist.ini, where port is the value of Application Server Control port in this file.

Compile RMS Oracle Forms and Reports

Note: This section details the compilation of the RMS 11 Forms 9i modules, using the compilation utilities provided within Oracle 9iAS 10g Forms and Reports Services. Forms Builder 9i is not included in Oracle 9iAS 10g Forms and Reports Services or Enterprise. In order to use Forms Builder 9i for manual compilation of RMS 11 Forms 9i modules, Oracle 9i Developer Suite (9iDS) 10g must be used. It should be noted that Oracle has not released a version of 9iDS for the AIX 5L platform. Appendix F details manual compilation steps using Oracle 9iDS 10g; Oracle 9iDS 10g installation instructions are not provided in this document.

Note: Because of Oracle bugs #3083648 and #2710859, and related Sun JVM Bug #4486745, some column headers and other font fields throughout the RMS application appear to have the bottom portion of text cut off when running in Forms 9i mode. This issue does not affect functionality. An Oracle enhancement request (3083648) has been made for a fix to the forms font handling mechanism.

Note: TNS must be set up within Oracle 9iAS 10g Forms and Reports Services in order to connect to the RMS 11 schema that was created in Chapter 2. This requires that 9iAS10G_ORACLE_HOME/network/admin/tnsnames.ora contain an entry for the Oracle 9i database that was created in Chapter2 (where the RMS 11 schema resides). See Appendix C for a tnsnames.ora example.

Create Staging Directory for RMS application server files

1 Log into the application server as the retek user created in Chapter 2 and determine where the RMS application files will be installed. There should be a minimum of 500 MB disk space available for the application installation files.

2 Copy the file rms11appserver.zip from the CD/appserverunix directory to the newly created staging directory. This will be referred to as INSTALL_DIR for the remainder of this chapter.

3 Change directories to INSTALL_DIR and extract the file rms11appserver.zip.

4 Make sure all scripts in INSTALL_DIR/forms9i_scripts have at lease execute permissions for the retek user and its group (r-xr-x---).

Setup

1 As the retek user, set the DISPLAY variable to the IP address plus ":0.0" (ie: 10.1.1.1:0.0) of the machine that is being used to perform the compilation from.

2 As the retek user, set the following variables:

Note: INSTALL_DIR is the location where RMS 11 will be installed.

Note: 9iAS10G_ORACLE_HOME is the location where Oracle 9iAS 10g is installed.

ORACLE_HOME=9iAS10G_ORACLE_HOME

PATH=$ORACLE_HOME/bin:INSTALL_DIR/forms9i_scripts:$PATH

Solaris only:
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/jdk/jre/lib/sparc:$ORACLE_HOME/jdk/jre/lib/sparc/native_threads

HP-UX only:
SHLIB_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/lib:$ORACLE_HOME/jdk/jre/lib/PA_RISC:$ORACLE_HOME/jdk/jre/lib/PA_RISC/server

AIX only:
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/jdk/jre/lib

```
LIBPATH=$LD_LIBRARY_PATH
```

All:
CLASSPATH=$ORACLE_HOME/jlib/debugger.jar:$ORACLE_HOME/jlib/utj90.jar:$ORACLE_HOME/jlib/ewt3.jar:$ORACLE_HOME/jlib/share.jar

FORMS90_BUILDER_CLASSPATH=$CLASSPATH

FORMS90_PATH=INSTALL_DIR/toolset/bin:INSTALL_DIR/rms/forms/bin:$ORACLE_HOME/forms90

REPORTS_PATH=INSTALL_DIR/rms/reports/bin:$ORACLE_HOME/forms90

Solaris/AIX only:
UP=<RMS schema owner>/<RMS schema password>@<RMS database>

HP-UX only:
UP=<RMS schema owner>/<RMS schema password>\@<RMS database>

Note: For the UP variable setting, the RMS schema and RMS database were created in Chapter 2. TNS must be set up within Oracle 9iAS 10g Forms and Reports Services in order to connect to the RMS 11 schema that was created in Chapter 2. See step 4 of the "Configure Oracle 9iAS 10g for RMS 11" section in this chapter for more information on the tnsnames.ora file. Verify that TNS is set up correctly by using the UP variable to successfully log into the RMS 11 schema.

Example: /u00/oracle> sqlplus $UP

Toolset

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 f90plsqlconv_pll_stand45 to automatically attach the Forms 9i 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 f90plsqlconv_pll_stand45.

5 Run pll2plx9i_toolset to compile all Toolset .pll's.

6 Check to make sure that each .pll file has a corresponding .plx (to ensure that all .pll's compiled successfully). If a library fails to compile (there is no .plx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions

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 fmb2fmx9i_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 fmb2fmx9i (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. If a form fails to compile (there is no .fmx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions.

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

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

14 Copy all menus (*.mmb files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory.

15 Run mmb2mmx9i (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. If a menu fails to compile (there is no .mmx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions.

Note: Should .err files 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.

Forms

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 pll2plx9i_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). If a library fails to compile (there is no .plx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions

5 Remove all newly created .plx files.

6 Copy all forms (*.fmb files) in the INSTALL_DIR/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory.

7 Run fmb2fmx9i_fm (in INSTALL_DIR/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 fmb2fmx9i (in INSTALL_DIR/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. If a form fails to compile (there is no .fmx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions.

Note: Disregard fm_*.fmx files should they be created. These files should be removed. They should NOT be copied to the INSTALL_DIR/rms/forms/bin directory.

11 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
```

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

13 Run mmb2mmx9i (in INSTALL_DIR/rms/forms/bin) to generate RMS runtime menus – .mmx's.

14 Check to make sure that each .mmb file has a corresponding .mmx file. If a form fails to compile (there is no .mmx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions.

15 Remove all .mmb files from INSTALL_DIR/rms/forms/bin.

Note: Should .err files be created by the compilation scripts above, these files are logs of the compilation process and can be removed.

Reports

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 pll2plx9i_reports to compile rep25lib.pll. If rep25lib.pll fails to compile (there is no .plx file), it will have to be manually compiled with Oracle 9iDS 10g. See Appendix F for manual compilation instructions

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 rdf2rep9i (in INSTALL_DIR/rms/reports/bin) to generate Reports runtime reports – .rep's.

Note: The following error messages may appear when running rdf2rep9i; 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. If a report fails to compile (there is no .rep file), it will have to be manually compiled with Reports Builder in Oracle 9iDS 10g. See Appendix F for manual compilation instructions.

8 Remove all .rdf files from INSTALL_DIR/rms/reports/bin.

Note: Should .err files be created by the compilation scripts above, these files are logs of the compilation process and can be removed.

Configure Oracle 9iAS 10g for RMS 11

Note: The proper Oracle 9iAS 10g components must be started in order to run Oracle Forms applications.

Note: 9iAS10G_ORACLE_HOME refers to the location where Oracle 9iAS 10g Forms and Reports Services is installed.

Note: Prior to modifying Oracle 9iAS 10g files, a backup of original files should be made.

1 Make a copy of the file 9iAS10G_ORACLE_HOME/forms90/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 FORMS90_PATH variable setting, and by adding the NLS_DATE_FORMAT variable to the end of this file. Additionally, the variable FORMS90_REJECT_GO_DISABLED_ITEM=FALSE must also be added to rms.env due to changes between Oracle Forms 6i and Oracle Forms 9i.

Example:
FORMS90_PATH=/u00/rms/toolset/bin:/u00/rms/forms/bin:/u00/oracle/9iAS10G/forms90

NLS_DATE_FORMAT=DD-MON-RR

FORMS90_REJECT_GO_DISABLED_ITEM=FALSE

3 A Reports Server needs to be running in order to access RMS 11 reports through the RMS 11 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 9iAS 10g Forms and Reports Services installation; at that time a 9i reports server entry was automatically made in 9iAS10G_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 9iAS10G_ORACLE_HOME/bin/rwserver.sh; all variables required for compiling 9i reports must be set, and there must be a 9i reports server entry in 9iAS10G_ORACLE_HOME/network/admin/tnsnames.ora prior to running rwserver.sh. In order to make a new 9i reports server entry in tnsnames.ora, make a copy of the default 9i reports server entry, changing the name and port (un-used port on the server). rwserver.sh must be run specifying the 9i reports server entry in the tnsnames.ora file.

Example (tnsnames.ora)**:** REP_RMS11 =
(ADDRESS = (PROTOCOL = tcp)(HOST = server)(PORT = 1951))

Example: rwserver.sh server=REP_RMS11

Verify the following resulted from running rwserver.sh:

- 9i reports server process started for the reports server specified
- configuration file was created in 9iAS10G_ORACLE_HOME/reports/conf for the reports server specified (named REP_RMS11.conf)

Note: Contact Oracle Support for problems with starting a 9i reports server. For disabling Single Sign On security with 9i reports, Oracle recommends removing the security tag from the REP_<SERVER_NAME>.conf file. Doing so can also solve problems with starting a 9i reports server.

4 Modify the file 9iAS10G_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

5 Make an entry in the file 9iAS10G_ORACLE_HOME/network/admin/tnsnames.ora for the Oracle 9i database that was created in Chapter2 (where the RMS 11 schema resides). Appendix C contains a sample tnsnames.ora file entry for an Oracle 9i database; refer to the sample or following example for a proper entry in file 9iAS10G_ORACLE_HOME/network/admin/tnsnames.ora.

6 Log into sqlplus as the RMS 11 schema owner (RMS11DEV) and update the lang table so that WEBHELP_SERVER, REPORTS_SERVER, WEBREPORTS_SERVER, and APP_SERVER are correct:

- WEBHELP_SERVER is the url **Error! Hyperlink reference not valid.** where <server> is the name or IP address of the server where Oracle 9iAS 10g is installed and <port> is the “Listen” value in 9iAS10G_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 **Error! Hyperlink reference not valid.** where <server> is the name or IP address of the server where Oracle 9iAS 10g is installed and <port> is the “Listen” value in 9iAS10G_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;

7 Modify the file formsweb.cfg located at 9iAS10G_ORACLE_HOME/forms90/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

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 will direct 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 9iAS10G_ORACLE_HOME/forms90/java/oracle/forms/registry/Registry.dat by setting default.icons.iconpath to /web_gif/.

Example: default.icons.iconpath=/web_gif/

9 Copy the RMS keyboard-mapping file INSTALL_DIR/sample_files/fmrweb.res to 9iAS10G_ORACLE_HOME/forms90/admin/resource/US.

10 Copy the sample file INSTALL_DIR/sample_files/rms11unix.conf to 9iAS10G_ORACLE_HOME/Apache/Apache/conf. rms11unix.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 9iAS 10g.

11 In rms11unix.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 rms11unix.conf to the end of httpd.conf, or add an include directive in httpd.conf to rms11unix.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 9iAS 10G Forms and Reports Services installation. The default OEM url should be http://server:1810.

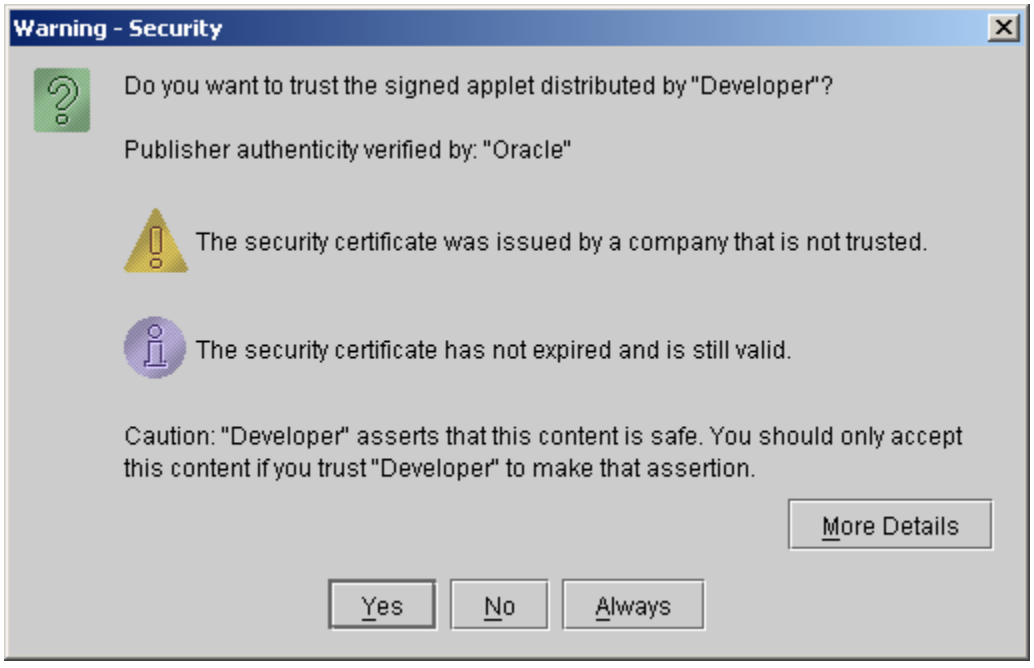
14 Load RMS in Forms 9i mode by entering the following url in a browser. Prior to testing, the Sun JRE 1.4.1+ plug-in needs to be installed on the client machine. The plug-in can be downloaded from http://java.sun.com/.

Error! Hyperlink reference not valid.

- server = name or IP address of server where Oracle 9iAS 10g is running
- port = Value of the "Listen" setting in 9iAS10G_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/forms90/f90servlet?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: RMS11DEV

Password: retek

Connect String: prod_db1

Appendix A – Generate DDL for Partitioned Tables

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/rms11_part.tab. This file will by used later 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 ../rms110.tab.
# The partitioned DDL file that will be generated is ../rms110_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
../rms11_part.tab file.
Completed successfully
```

Appendix B – Oracle 9.2.0.x Database Creation Scripts

```
###############################################################################
# Oracle 9.2.0.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
# +------+ +---------+ +-----------------+ +-------------+ +-----------
--+
# 07/01/04 Retek      NA                      NA              creation
#
###############################################################################

# ------------------------------------------------------------------------
# The following SGA parameters are CRITICAL to the performance of the
# database. The following settings are based off 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                  = 24M                 #150M or higher
for applying oracle patchsets; 70M if using RMAN
log_buffer                      = 10485760
shared_pool_size                = 150M

# ------------------------------------------------------------------------
# The following parameters do not affect SGA size
# ------------------------------------------------------------------------
audit_file_dest                 = <admin_path>/adump
background_dump_dest            = <admin_path>/bdump
compatible                      = 9.2.0
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_files                        = 999                    # default is
200; set to max number of database files
db_file_multiblock_read_count   = 16                      # (max io
size)/(block size); adjust as needed; platform specific

db_name                         = SID
db_writer_processes             = 4
```

```
job_queue_processes             = 9                     # Retek
required; number of cpu's +1
local_listener                  =
"(ADDRESS=(PROTOCOL=TCP)(HOST=<hostname>)(PORT=1521))"
nls_date_format                 = DD-MON-RR             # Retek
required
nls_calendar                    = GREGORIAN
nls_language                    = AMERICAN              # default
nls_territory                   = AMERICA               # default
open_cursors                    = 900                   # Retek
required (minimum=900); default is 50
optimizer_features_enable       = 9.2.0
optimizer_mode                  = CHOOSE                # Retek
required
processes                       = 500                   # max number of
os processes that can connect to oracle
query_rewrite_enabled           = TRUE                  # fct based
indexes
sessions                        = 1500                  #
~(3*processes);
session_cached_cursors          = 100                   # default is 0
undo_management                 = AUTO
undo_tablespace                 = undo_ts
undo_retention                  = 1800                  # currently set
for 30 minutes; set to avg length of transactions in secs
user_dump_dest                  = <admin_path>/udump
utl_file_dir                    = <utl_file_path>

# ***  Archive Logging, set if needed  ***
#log_archive_dest_1               = 'location=<admin_path>/arch/'
#log_archive_format              = SIDarch%s.log
#log_archive_min_succeed_dest    = 1
#log_archive_start               = TRUE
#log_checkpoint_interval         = 9999999999
```

```
-------------------------------------------------------------------------
--- Script:      crdb1.sql
--- Execute as:  sysdba
--- Note:        Before running this script:
---                 Modify <datafile_path> values.
---                 Modify SID values.
---                 Adjust sizes for redo logs, TEMP and UNDO
--                  tablespaces.
-------------------------------------------------------------------------
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 10M 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
;

spool off
exit
```

```
----------------------------------------------------------------------
--- Script:        crdb2.sql
--- Execute as:    sysdba
--- Note:          This script installs the data dictionary views
----------------------------------------------------------------------
spool crdb2.log

REM # install data dictionary views:
PROMPT Running catalog.sql
@$ORACLE_HOME/rdbms/admin/catalog.sql
PROMPT Running catproc.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql
PROMPT Running catblock.sql
@$ORACLE_HOME/rdbms/admin/catblock.sql
PROMPT Running profload.sql
@$ORACLE_HOME/rdbms/admin/profload.sql


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 select on dba_sys_privs to public with grant option;
grant execute on dbms_rls to public with grant option;

REM * These privs needed to be granted to all due to 9i security
REM changes.
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;

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;

connect system/manager
PROMPT Running pupbld.sql
@$ORACLE_HOME/sqlplus/admin/pupbld.sql

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;

disconnect

exit
```

```
--------------------------------------------------------------------------
--- Script:        crdb3.sql
--- Execute as:    sysdba
--- Note:          This script installs java and xml components
--------------------------------------------------------------------------
spool crdb3.log

REM *  Install XDK and XSU
PROMPT altering system to set _system_trig_enabled to false
ALTER SYSTEM SET "_system_trig_enabled"=FALSE SCOPE=MEMORY;

PROMPT Running initjvm.sql to install Java objects
@$ORACLE_HOME/javavm/install/initjvm.sql

PROMPT Running initxml.sql to install XML and XSU
@$ORACLE_HOME/rdbms/admin/initxml.sql

PROMPT Running xmlja.sql to install NCOMP'ed XML Parser
@$ORACLE_HOME/xdk/admin/xmlja.sql

PROMPT Running catjava.sql to install catalog scripts for Java
@$ORACLE_HOME/rdbms/admin/catjava.sql

PROMPT Creating public synonyms and grants
CREATE PUBLIC SYNONYM XMLQUERY for SYS.DBMS_XMLQUERY;
GRANT EXECUTE ON XMLQUERY TO PUBLIC;

GRANT EXECUTE ON XMLPARSER TO PUBLIC;
GRANT EXECUTE ON XMLDOM TO PUBLIC;
CREATE PUBLIC SYNONYM XSLPROCESSOR for SYS.XSLPROCESSOR;
GRANT EXECUTE ON XSLPROCESSOR TO PUBLIC;

PROMPT Revaliding invalid objects
@$ORACLE_HOME/rdbms/admin/utlrp.sql

spool off

exit
```

Appendix C – 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)))
```

Appendix D – 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
```

Appendix E – Retek 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.

```
create user schema_owner
identified by retek
default tablespace RETEK_DATA
temporary tablespace temp;
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,
      query rewrite,
      create materialized view to &schema_owner
/
alter user schema_owner quota unlimited on retek_data
/
alter user schema_owner quota unlimited on retek_index
```

```
/
alter user schema_owner quota unlimited on lob_data
/
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
/
```

Appendix F – Manual RMS Installation Instructions

Compile RMS Oracle Forms and Reports

Note: This section details the compilation of the RMS 11 Forms 9i modules using Oracle 10G Forms and Reports Services(Forms 9i version 9.0.4).

Setup

1 Log into the application server as the retek user created in Chapter 2.

2 As the retek user, set the DISPLAY variable to the IP address of the machine that is being used to perform the compilation from.

3 As the retek user, set the following variables:

Note: INSTALL_DIR is the location where RMS 11 will be installed.

Note: 9iAS10G_ORACLE_HOME is the location where Oracle 10g is installed.

ORACLE_HOME=9iAS10G_ORACLE_HOME

PATH=$ORACLE_HOME/bin:INSTALL_DIR/form9i_scripts:$PATH

Solaris only:
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/jdk/jre/lib/sparc:$ORACLE_HOME/jdk/jre/lib/sparc/native_threads

HP-UX only:
SHLIB_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/lib:$ORACLE_HOME/jdk/jre/lib/PA_RISC:$ORACLE_HOME/jdk/jre/lib/PA_RISC/server

CLASSPATH=$ORACLE_HOME/jlib/debugger.jar:$ORACLE_HOME/jlib/utj90.jar:$ORACLE_HOME/jlib/ewt3.jar:$ORACLE_HOME/jlib/share.jar

FORMS90_BUILDER_CLASSPATH=$CLASSPATH

FORMS90_PATH=INSTALL_DIR/toolset/bin:INSTALL_DIR/rms/forms/bin:$ORACLE_HOME/forms90

REPORTS_PATH=INSTALL_DIR/rms/reports/bin:$ORACLE_HOME/forms90

UP=<RMS schema owner>/<RMS schema password>@<RMS database>

HP-UX only:
UP=<RMS schema owner>/<RMS schema password>\@<RMS database>

Note: For the UP variable setting, the RMS schema and RMS database were created in Chapter 2. TNS must be set up within Oracle 10g Forms and Reports Services in order to connect to the RMS 11 schema that was created in Chapter 2. See step 4 of the "Configure Oracle 9iAS 10g for RMS 11" section in this chapter for more information on the tnsnames.ora file. Verify that TNS is set up correctly by using the UP variable to successfully log into the RMS 11 schema.

Example: sqlplus $UP

Compile Toolset Libraries (*.pll)

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

```
> cp *.pll ../bin
```

2 Change directories to INSTALL_DIR/toolset/bin.

3 Run f90plsqlconv_pll_stand45 to automatically attach the Forms 9i library rp2rro.pll to stand45.pll. This library must be attached to stand45.pll in order to run RMS reports.

```
> f90plsqlconv_pll_stand45
```

4 Remove the newly created stand45.pld should it be created from running f90plsqlconv_pll_stand45

5 Start Form Builder 9i to compile all Toolset libraries manually.

```
> f90desm &
```

a A blue GUI interface is displayed. On the Welcome page click Cancel.

b Chose File -> Connect. Log into the database as the RMS 11.x schema owner.

c Compile the libraries in the following order:

- 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

For each library file:

a Choose File -> Open.

b Select INSTALL_DIR/toolset/bin/FILENAME.pll

c Click OK.

d Once the library is “loaded”, select the library name, select Program, and choose Compile -> All.

e After successful compilation, click OK.

f Save and close the library.

The Toolset libraries have been compiled and should now reside in the bin directory.

Compile Toolset Reference Forms (fm_*.fmb)

1 Copy all reference forms (fm_*.fmb files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory.

```
> cp fm_*.fmb ../bin
```

2 Change directories to INSTALL_DIR/toolset/bin.

3 Using Form Builder session (f90desm &), navigate to INSTALL_DIR/toolset/bin and compile each Toolset reference form in the INSTALL_DIR/toolset/bin directory:

a Choose File -> Open.

b Select INSTALL_DIR/toolset/bin/fm_FILENAME.fmb.

c Click OK.

d Once the reference form is “loaded”, select the form name, select Program, and choose Compile -> All.

e After successful compilation, click OK.

f Save and close the reference form.

The Toolset reference forms have been compiled and should now reside in the bin directory.

Compile Toolset Forms (*.fmb)

1 Copy all non-reference form forms (*.fmb files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory. Do NOT copy fm_*fmb files from the /src to /bin directory

Note: Make sure not to copy reference forms (fm_*.fmb files) again from the /src to /bin directory as this will overwrite the reference forms that were compiled in step 8 above and possible cause compilation errors for the remaining modules.

2 Change directories to INSTALL_DIR/toolset/bin.

3 Run fmb2fmx9i to generate Toolset runtime forms – .fmx’s.

```
> fmb2fmx9i
```

4 Check to make sure each non-reference form .fmb file has a corresponding .fmx file. If a form fails to generate (there is no .fmx file), may have to manually compile the form in Form Builder session (f90desm &).

   a Choose File -> Open.

   b Select INSTALL_DIR/toolset/bin/FILENAME.fmb.

   c Click OK.

   d Once the form is "loaded", select the form name, select Program, and choose Compile -> All.

   e After successful compilation, click OK.

   f Press 'Ctrl + T' to convert the form to binary mode.

   g Save and close the form.

5 Remove non-reference form forms from INSTALL_DIR/toolset/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

The Toolset runtime forms have been generated and should now reside in the bin directory.

Compile Toolset Menus (*.mmb)

1 Copy all menus (*.mmb files) in the INSTALL_DIR/toolset/src directory to the INSTALL_DIR/toolset/bin directory.

```
> cp *.mmb ../bin
```

2 Change directories to INSTALL_DIR/toolset/bin.

3 Run mmb2mmx9i to generate Toolset runtime menus – .mmx's.

```
> mmb2mmx9i
```

4 Check to make sure each .mmb file has a corresponding .mmx file. If a menu fails to generate (there is no .mmx file), may have to manually compile/generate the menu in Form Builder session (f90desm &).

   a Choose File -> Open.

   b Select INSTALL_DIR/toolset/bin/FILENAME.mmb.

   c Click OK.

   d Once the menu is "loaded", select the menu name, select Program, and choose Compile -> All.

   e After successful compilation, click OK.

   f Press 'Ctrl + T' to convert the menu to binary mode.

   g Save and close the menu.

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

```
> rm *.mmb
```

The Toolset runtime menus have been generated and should now reside in the bin directory.

Compile RMS Forms Libraries (*.pll)

1 Copy all libraries (.pll files) in the INSTALL_DIR/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory.

```
> cp *.pll ../bin
```

2 Change directories to INSTALL_DIR/rms/forms/bin.

3 Using Form Builder session (f90desm &), navigate to INSTALL_DIR/rms/forms/bin and compile each RMS library in that directory, in the following order:

- links45.pll
- itemuda.pll
- og.pll

For each library file:

a Choose File -> Open

b Select INSTALL_DIR/rms/forms/bin/<FILENAME>.pll.

c Click OK

d Once the library is “loaded” select the form library name, select Program, and choose Compile -> All.

e After successful compilation click OK.

f Save and close the form library.

The RMS libraries have been compiled and should now reside in the bin directory.

Compile RMS Reference Forms (fm_*.fmb)

1 Copy all reference forms (fm_*.fmb files) in the INSTALL_DIR/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory.

```
> cp fm_*.fmb ../bin
```

2 Change directories to INSTALL_DIR/rms/forms/bin.

3 Using Form Builder session (f90desm &), navigate to INSTALL_DIR/rms/forms/bin and compile each RMS reference form in that directory.

   a Choose File -> Open

   b Select INSTALL_DIR/rms/forms/bin/fm_<FILENAME>.fmb.

   c Click OK.

   d Once the reference form is "loaded" select the form name, select Program, and choose Compile -> All.

   e After successful compilation click OK.

   f Save and close the reference form.

The RMS reference forms have been compiled and should now reside in the bin directory.

Compile RMS Forms (*.fmb)

1 Copy all non-reference form forms (*.fmb files) in the INSTALL_DIR/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory. Do NOT copy fm_*fmb files from the /src to /bin directory

   Note: Make sure not to copy reference forms (fm_*.fmb files) again from the /src to /bin directory as this will overwrite the reference forms that were compiled in the previous section above and possible cause compilation errors for the remaining modules.

2 Change directories to INSTALL_DIR/rms/forms/bin.

3 Run fmb2fmx9i to generate RMS runtime forms – .fmx's.

```
> fmb2fmx9i
```

4 Check to make sure each .fmb file has a corresponding .fmx file. If a form fails to compile (there is no .fmx file), may have to manually compile the form in Form Builder session (f90desm &).

   a Choose File -> Open.

   b Select INSTALL_DIR/rms/forms/bin/FILENAME.fmb.

   c Click OK.

   d Once the form is "loaded", select the form name, select Program, and choose Compile -> All.

   e After successful compilation, click OK.

   f Press 'Ctrl + T' to convert the form to binary mode.

   g Save and close the form.

5 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
```

The RMS forms have been compiled and should now reside in the bin directory.

Compile RMS Menus (*.mmb)

1 Copy all menus (*.mmb files) in the INSTALL_DIR/rms/forms/src directory to the INSTALL_DIR/rms/forms/bin directory.

```
> cp *.mmb ../bin
```

2 Change directories to INSTALL_DIR/rms/forms/bin.

3 Run mmb2fmx9i to generate RMS runtime menus – .mmx's.

```
> mmb2mmx9i
```

4 Check to make sure each .mmb file has a corresponding .mmx file. If a menu fails to compile (there is no .mmx file), may have to manually compile the menu in Form Builder session (f90desm &).

   a Choose File -> Open.

   b Select INSTALL_DIR/rms/forms/bin/FILENAME.mmb.

   c Click OK.

   d Once the menu is "loaded", select the form name, select Program, and choose Compile -> All.

   e After successful compilation, click OK.

   f Press 'Ctrl + T' to convert the menu to binary mode.

   g Save and close the menu.

5 Remove all .mmb files from INSTALL_DIR/rms/forms/bin

```
> rm *.mmb
```

The RMS menus have been compiled and should now reside in the bin directory.

Compile RMS Reports Library (rep25lib.pll)

1 Copy the reports library (rep25lib.pll) in the INSTALL_DIR/rms/reports/src directory to the INSTALL_DIR/rms/reports/bin directory.

```
> cp rep25lib.pll ../bin
```

2 Change directories to INSTALL_DIR/rms/reports/bin.

3 Using Form Builder session (f90desm &), navigate to INSTALL_DIR/rms/reports/bin and compile the RMS reports library in that directory.

   a Choose File -> Open.

   b Select INSTALL_DIR/rms/reports/bin/rep25lib.pll.

   c Click OK.

   d Once rep25lib.pll is "loaded" select the library, select Program, and choose Compile -> All.

   e After successful compilation click OK.

   f Save and close rep25lib.pll

The RMS reports library has been compiled and should now reside in the bin directory.

Compile Reports (*.rdf)

1 Copy all reports (*.rdf files) from the INSTALL_DIR/rms/reports/src directory to the INSTALL_DIR/rms/reports/bin directory

```
> cp *.rdf  ../bin
```

2 Run rdf2rep9i (in INSTALL_DIR/rms/reports/bin) to generate RMS reports – .rep's.

```
> rdf2rep9i
```

Note: The following error messages may appear when running rdf2rep9i; 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

3 Check to make sure each .rdf file has a corresponding .rep file. If a report fails to compile (there is no .rep file), may have to manually compile the report with Reports Builder 9i. If necessary, start Reports Builder 9i to compile reports manually.

```
> ./rwbuilder.sh &
```

a A blue GUI interface is displayed. On the Welcome page click Cancel.

b Chose File -> Connect. Log into the database as the RMS 11.x schema owner.

c Choose File -> Open.

b Select INSTALL_DIR/rms/reports/src/FILENAME.rdf.

c Click OK.

d Once the report is “loaded”, select the report name, select Program, and choose Compile -> All.

e After successful compilation, click OK.

f Press ‘Ctrl + T’ to convert the form to binary mode.

g Save and close the report.

4 Remove all .rdf files from INSTALL_DIR/rms/reports/bin.

```
> rm *.rdf
```

The RMS reports have been compiled and should now reside in the bin directory.

Appendix G – AIX Shared Library Bug Fix

The make files for Oracle 9i have minor bugs in them. The following changes in **bold** need to be made to the $ORACLE_HOME/rdbms/lib/env_rdbms.mk file. Similar changes should be made to the BUILDLIB_NO_CONTEXT function as well.

```
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)
```

TAR # 1733664.995 was logged with Oracle and bug #2143531 was filed but was not fixed because there is a workaround.

Appendix H – 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 will read data, create, delete and update 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). need to set MMHOME to base directory for RMS ETL. This is the top level directory that selected during the installation process. So in .kshrc should add a line like the following:

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

rmse_config.env

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

```
export DBNAME=int9i
export RMS_OWNER=RMS11DEV
export BA_OWNER=rmsint1011
```

Also, 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=retek`

Appendix I – Retek Trade Management 11 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 will insert the Expense and Assessment Cost Components. This script will need 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 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 will 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 I 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 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. 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 will contain 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 will be 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 will be 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 will 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 will 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 will need 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 will error 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 will be defaulted in along with all other assessments that are associated with each Order/Item combination. Once associated with the Entry, MPF will be recalculated and checked to see if the value falls within the Min/Max Range. If not, the value will be 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 will not be 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 will require Unit of Measure (UOM) conversions in multiple instances. If a particular UOM conversion is missing the processing will stop 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 will be in ‘Worksheet’ status. Once all of the necessary information has been added, the user will 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 will be downloaded to the Broker (cednld.pc). Once the download process is complete, the Status will automatically be 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 will remain editable. All information on the CE Shipment form will be view only. Also, all information on the CE Order/Item form will be 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.