Oracle® Retail Data Warehouse Database Installation Guide

Release 12.0

May 2006

Contents

Preface

Oracle Retail Installation Guides contain the requirements and procedures that are necessary for the retailer to install Oracle Retail products.

Audience

This Installation Guide is written for the following audiences:

- Database administrators (DBA)
- System analysts and designers
- Integrators and implementation staff

Related Documents

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

- Oracle Retail Data Warehouse Data Model
- Oracle Retail Data Warehouse Middle Tier Installation Guide
- Oracle Retail Data Warehouse Operations Guide
- Oracle Retail Data Warehouse Release Notes
- Oracle Retail Data Warehouse User Guide
- MicroStrategy documentation

Customer Support

- https://metalink.oracle.com

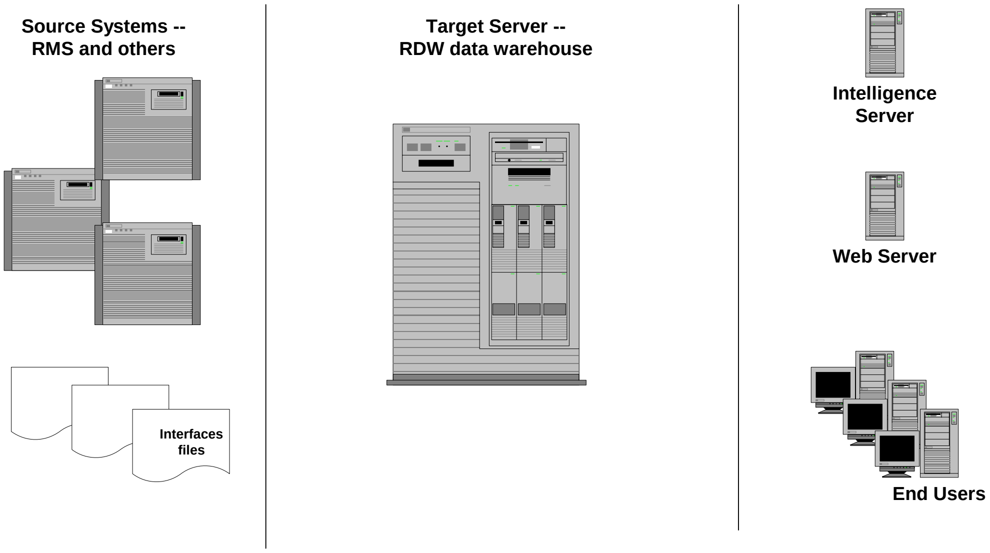
When contacting Customer Support, please provide:

- 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.

1
Pre-Installation Tasks

This release of RDW incorporates optional interfaces with these Oracle Retail products which can be sources for the data warehouse: Oracle Retail Merchandising System (RMS), Oracle Retail Invoice Matching (ReIM), and Oracle Retail Price Management (RPM). Additionally, the data warehouse can also operate as a standalone product and be fed from other legacy systems. If Oracle Retail applications are used as the source systems, follow the requirements in the installation guides for each of these applications. It is recommended that the source systems be on a separate server from the data warehouse which is considered the target server. Refer to Appendix A for more detailed information on RDW and database schema setup.

RDW Architecture on Oracle



Check Database Server Hardware Requirements

Hardware requirements for the data warehouse database server or target server include:

General Hardware Requirements	Minimum	Recommended
Memory	8GB	12GB+
Multi-processors	4 – highest MHz	8+ – highest MHz
CD-ROM drive (either local or network)		
Disk Space	300 GB	Site specific (refer to your RDW sizing information)
Swap Space	Equal to physical RAM	2.5 times the physical RAM

Note: This reflects the amount of space needed to install the database software, RETL software, as well as the RDW software, and still maintain an acceptable amount of usage on the Unix file system. This may reflect external storage as well as internal, such as 50GB of internal space and a 250+ GB external RAID storage array.

Sizing Factors and other suggestions to factor into the selection of a data warehouse server include:

- Concurrent front-end user base. More concurrent users will require more database processing power from the server.
- DASD connectivity. Typically, it is better if the fiber channel connectivity to provides the maximum throughput to disk.
- Backup/Recovery requirements. Extra disks may be required for backup and recovery procedures.
- Server backplane speeds. Depending on the architecture of the server backplane (or comparable links), memory access and/or CPU utilization may be a factor in performance.
- Overall database size for the RDW. A capacity plan should be done for the database server and DASD requirements in general to assess how large the environment will need to be.

Note: These data warehouse server requirements should only be used as guidelines because they reflect the hardware used to run a small environment (approximately 200 gigabytes of data). Actual requirements can be somewhat smaller, or typically much larger, depending on the intended size of the RDW environment upon full implementation. Oracle can assist with making these determinations by providing information on database size estimates, server architecture, and so on.

Check Database Server Software Requirements

General Requirements for a database server running RDW include:

- UNIX based OS certified with Oracle RDBMS 10g Enterprise Edition (options are AIX5.2, AIX5.3, Solaris 9, and HP-UX 11.11)
- Oracle RDBMS 10g Release 2 Enterprise Edition (minimum 10.2.0.1.0 patchset required) with the following patches and components:

  Patches:

  - 4516865 (WRONG PERMISSIONS AFTER INSTALLATION IN OH AND SUBSEQUENT DIRECTORIES)

  Components:

  - Oracle Database 10g
  - Oracle Partitioning
  - Oracle Net Services
  - Oracle Call Interface (OCI)
  - Oracle Programmer
  - Oracle XML Development Kit
- Perl compiler 5.0 or later
- ANSI compliant C compiler (certified with OS and database version)
- x-Windows interface
- Other software requirements for the data warehouse include: Please refer to the *RDW 12.0 Middle Tier Installation Guide* for the full installation instructions of the MicroStrategy components.
  - MicroStrategy Intelligence Server 8.0.2
  - MicroStrategy Web 8.0.2
  - MicroStrategy Desktop 8.0.2
  - MicroStrategy Administrator 8.0.2
  - Oracle Retail Extract Transform and Load (RETL) 12
- Retek product interfaces to the data warehouse include: Please refer to Chapter 6 – RDW Interfaces of the RDW 12.0 Operations Guide for mode details about each interface.
  - RMS 12.0
  - ReIM 12.0
  - RPM 12.0

Check Client System Requirements

General requirements for end-user client workstation software requirements include:

- See specific database platform documentation for client requirements.
- See the Retek Data Warehouse Middle Tier Installation Guide for other client requirements.

Create Unix User Accounts

1. Log in as the `root` user.
2. Create the following groups:
   - dba - admin group which controls RDW access
   - dev - development group
3. Create the following user accounts, using `ksh` as the default shell:
   - oretail - dba group
   - rdw12dev - dev group
4. Typically the owner of the Oracle RDBMS software will own the RDW database software. This user will be referred to as *`rdwdba`*.

Create Unix Directories

1. Create Unix directories for the RDW database structure and the code directory structure.

   Note: The database directories are used to provide scripts to aid in the creation of tablespaces and users. The code directory structure holds the source code, error and log directories. Refer to Appendix B: Code Tree for a more detailed description of the directory structure.

2. Create RDW database directory structure:
   - Log in to the database server as UNIX user `rdwdba`. This user will own the database directory structure.
   - Extract the files from the downloaded zip file. FTP (in binary mode) the tar file, rdw12_db.zip, to <database_directory> on the Unix server where the rdw12 database will be created. The <database_directory> directory is a working directory that should be designated at this time. As the `rdwdba` user, extract the rdw12_db.zip file by typing the following at the unix prompt:

     ```
     unzip rdw12_db.zip
     ```

3. Create RDW code directory structure:
   - Log in to the database server as the UNIX user oretail. This user will own the code directory structure.
   - FTP (in binary mode) the file `rdw12_code.zip` to a base directory where you wish to permanently store the RDW code directory. This base directory should be on the Unix server where the RDW 12.0 database will be created. The base directory is a working directory that you should designate at this time. As the oretail user, extract the `rdw12_code.zip` file in the base directory specified above by entering the following at the unix prompt:

     ```
     unzip rdw12_code.zip
     ```

     Note: It is highly recommends that clients have a separate server for production installations. To set up a production environment, create a prd directory, then copy the dev directory structure to the production server and use the `<base_directory>/rdw12.0/prd` as the code directory structure on that server. A prd group is needed and a separate prd user should be created as well.

Alter Directory Ownership and Privileges

For added security, alter the directory ownership and privileges.

1. Log in as the `root` user (in the `<base_directory>/rdw12.0 directory`).
2. Change the following as indicated in the table below:
   - the ownership `(chown –R <owner>  <directory>)`
   - the group `(chgrp –R <user group>  <directory>)`
   - the privileges `(chmod –R <privs>  <directory>)`

Directory	Owner	Group	Privilege
retek	oretail	dba	775
dev	rdw12dev	dev	775
dbasql	rdwdba – Oracle	dba	774

For example: cd <*base_directory*>/rdw12.0
chown –R rdw12dev dev
chgrp –R dev dev
chmod –R 775 dev

Copy profiles for RDW User

1. Login to UNIX as the rdw12dev user and change directories to `<base_directory>/rdw12.0/sample_profiles`.
2. Copy the profile to the home directory of rdw12dev. Edit the .profile to correspond with the server file system, RETL installation and Unix environment.

Note: The sample profiles by default are set up for a development environment. Change the indicated variables as necessary for other environments. Any variable that may require modification are noted in the `sample profile` comments. Be sure to set up these Unix accounts to automatically run the `.profile` within their home directory upon login.

2 Installation Tasks

It is assumed that Oracle 10g release 2, with appropriate patches, has already been installed. If not, refer to “Check Database Server Software Requirements” in Chapter 1, “Pre-Installation Tasks” before proceeding.

Note: Become familiar with the RDW application in a development environment before setting up a production system. The following instructions are recommended for development and test environments only. When implementing RDW for a production environment, refer to capacity planning information to determine size requirements for tablespaces, tables, and indexes. The installation scripts provided will need to be modified accordingly.

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

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

Establish Data Warehouse Partitioning Strategy

Establish a partitioning strategy before creating compressed datamart and historical tables in a production environment. In doing so, consider the database size and business requirements. For example, the amount of history to be held at various levels, and the various functional areas that might be used should be referenced when determining a partitioning strategy. Additionally, large non-compressed fact tables can be should be partitioned for ease of rolling off history. Refer to the RDW Operations Guide for more detailed information regarding the partitioning strategy for both compressed and non-compressed fact tables. Refer to “Oracle® Database Concepts 10*g* Release 2 (10.2) Chapter 18 “Partition Tables and Indexes” for further details regarding partitioning concepts.

Create the RDW Database

Create the Database as Follows

1. Login to the RDW database server as the *rdwdba* user
2. Create the Oracle recommended OFA directory structure for the database (datafile directories, adump, bdump, cdump, arch, create, exp, pfile, udump, utl_file_dir)
3. Place an entry in the oratab file for the database and execute oraenv to set the ORACLE_SID and ORACLE_HOME environment variables.
4. Copy INSTALL_DIR/rdw12/create_db/init.ora to the $ORACLE_HOME/pfile directory and rename it to init${ORACLE_SID}.ora. Modify the parameters according to guidelines specified in this file.
5. Create a symbolic link from $ORACLE_HOME/pfile/init${ORACLE_SID}.ora to $ORACLE_HOME/dbs/init${ORACLE_SID}.ora.
6. Modify the INSTALL_DIR/rdw12/create_db/crdb1.sql file. Refer to comments in this file regarding modifications that need to be made.
7. Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/rdw12/create_db/crdb1.sql. Review crdb1.log for errors and correct as needed.
8. Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/rdw12/create_db/crdb2.sql. Review crdb2.log for errors and correct as needed.
9. Login to SQL*Plus as SYSDBA and execute INSTALL_DIR/rdw12/create_db/crdb3.sql. Review JServer.log, context.log and xdb_protocol.log for errors and correct as needed.
10. Configure the database listener.

Create RDW Tablespaces

1. As the rdwdba user, change directories to INSTALL_DIR/rdw12/create_db.
2. Modify the create_rdw_tablespaces.sql script as appropriate. Refer to Appendix D.
3. Login to SQL*Plus as SYSDBA and execute:

   ```
   SQL>@create_rdw_tablespaces.sql.
   ```

   - Review create_rdw_tablespaces.log for errors and correct as needed.
4. If a separate RMS database will be used for the data warehouse interface (dwi), run the following script in the RMS database (refer to Appendix D):

   ```
   SQL>@create_rdw_dwi_tablespace.sql.
   ```
5. Review create_rdw_dwi_tablespace.log for errors and correct as needed.

Load RDW Database

Create RDW Schema Owners

1. As the rdwdba user, change directories to INSTALL_DIR/rdw12/utility
2. Review the crusers.sql file for the names and passwords of the rdw users that will be created. Note these as they will be used later in the installation.
3. Login to SQL*Plus as SYSDBA and execute the following script:
   `SQL>@crusers..sql`
4. To secure the passwords in the create users script, change the privileges on the script by entering: `chmod 700 crusers.sql`

Create RDW Datamart-level tables

1. As the rdwdba user, change directories to `<database_directory>/rdw12/ddl/dm`.
2. Log in to SQL*Plus as `rdw12dm` and execute the following script:
3. `SQL>@rdw12.sql`
4. This script creates the RDW datamart-level tables. It will create all default base non-partitioned tables. If you are implementing a partitioning strategy at this time, modify the DDL to reflect your partitioning.
5. Review the `rdw12dm.log` file to verify that the objects were all created successfully.

Install MicroStrategy Metadata

In order to run RDW reports from MicroStrategy, a metadata layer is needed in the database. The metadata tables hold information about reports, filters, metrics, and so on. The tables for MicroStrategy have already been set up and exist in export files. The steps below outline how to install the MicroStrategy repository including all RDW reporting objects. This approach will install all objects with the Oracle import utility, which is faster for an initial installation of all four RDW Workbenches.

1. As the rdwdba user, change directories `to <database_directory>/rdw12/ddl/md.`
2. Set NLS_LANG environment variable to NLS_LANG =AMERICAN_AMERICA.UTF8.
3. The script, import_rdw12md.sh will import the metadata objects using an export file. At a Unix prompt enter: ./import_rdw12md.sh
   - `At the prompt enter the username where the metadata should be created, rdw12md.`
   - `At the prompt choose the appropriate language for your implementation.`
   - `Enter the username and password to execute the import. It is recommended that sys or system` be used.
   - Review the rdw12md.log file for errors and make appropriate corrections if needed and rerun.

Set up RDW User Permissions

1. As the rdwdba user, change directories on the Unix server to `<base_directory>/rdw12.0/dbasql`.
   - This directory contains scripts that will set up all necessary synonyms and grants for the RDW environment. The header of each file contains information regarding which user should execute the script, when to run them and additional details about the scripts.
   - The scripts will spool to the `/tmp` directory, which should already exist, but verify that this directory exists and can be written to before running.
   - The rdw_users.sql script has been set up to reference the default RDW usernames, so verify that they are correct and change to suit your environment if necessary.
   - When prompted by any of the scripts for an object name, press the **[Enter]** key to create permissions for all objects in the schema.
2. Login to SQL*Plus as the schema owner `rdw12dm (created earlier)` and execute the following scripts to setup permissions to the datamart objects:

   ```
   SQL>@gs_rdw12_dm_ora.sql
   SQL>@invalids.sql
   ```

   - Verify the scripts executed correctly by reviewing the following files: `g_rdw12_dm.lst, s_rdw12_dm.lst`
3. Login to SQL*Plus as the schema owner as `rdw12md (created earlier)` and execute the following scripts to setup permissions to the MicroStrategy database objects:

   ```
   SQL>@gs_rdw12_md_ora.sql
   SQL>@invalids.sql
   ```

   - Verify the scripts executed correctly by reviewing the following files: `g_rdw12_md.lst, s_rdw12_md.lst`

Note: Ignore any text that is generated similar to the following. `SP2-0734: unknown command beginning "new        7:  ..." - rest of line ignored`

Ignore any text that is generated from invalids.sql if no objects need to be validated.

```
SP2-0734: unknown command beginning "no rows
se..." - rest of line ignored.
```

Note: These scripts are also used for maintenance. If you need to drop or add a table in the database, run these scripts again to give the appropriate privileges to your users. If a table has been dropped and recreated for maintenance, you also need to verify that all objects in the database are VALID. For a test or a production environment, you need to modify the rdw_users.sql script to alter the usernames for the environment you are creating.

Configure RETL

This release of RDW incorporates the RETL. Take this time now to install and configure this on the target server. See the RETL Programmers Guide to install and configure this tool. Once configured, populate the default data as described in the rest of this chapter.

1. Log in to the Unix server with the `rdw12dev` account.
2. Change directories to `<base_directory>/rdw12.0/rfx/etc`.
3. Modify the rdw_config.env script to match your environment. For detail information about each setting, refer to section ‘rdw_config.env settings’ within the RDW Operation Guide.

Data Population

In order to successfully populate the database with datamart data, the rdw12dev profile must be modified for these values: `$MMUSER`, `$PASSWORD`, $PATH, $RFX_HOME and `$MMHOME`.

- Log in to a Unix session on the RDW database server as rdw12dev. Modify .profile by setting MMUSER to the data mart owner, rdw12dm. Set PASSWORD to the appropriate password as determined when the crusers.sql script was run.
- Set RFX_HOME to the location of the RETL installation and enter this value in the PATH.
- Source the profile or exit and log back on as rdw12dev.
- Verify the RETL executable is in the path of your Unix session by typing the following at the UNIX prompt:

  ```
  which rfx
  ```

Populate Datamart Default Data

1. As the rdw12dev user, change directories to `<base_directory>/rdw12.0/install`. This directory contains the code for all default data installation scripts.
2. At the Unix prompt, enter:
   `default_load.ksh`

   Note: Read the comments on every install scripts called within default_load.ksh specifically load_maint_dim_key_dm.sql because the initial date for curr_load_dt might be different for every client.

   This script executes the following SQL scripts:

   - alter_dt_fmt.sql - alters the current session's date format.
   - load_program_control_dm.sql - populates the program status data for the RDW modules.
   - load_dummyrows.sql - populates necessary 'dummy' records for low level dimensions.
   - load_maint_dim_key_dm.sql - populates the dimension surrogate key information and initial maintenance date.
   - load_indicator_dm.sql - populates the default values for a dummy table for front-end reporting.
   - load_cde_dtl_dm.sql - populates the code abbreviation table for RDW batch modules to reference.
   - load_prod_diff_type_dm.sql - populates the diff type dimension with default diff type values.
   - load_rtl_type_dm.sql - populates the retail type dimension table with default retail types.
   - load_cde_dtl_com_dm.sql- populates the customer order code abbreviation table for RDW batch modules to reference.
   - load_tsf_type_dm.sql – populates the transfer type dimension table with default transfer types.
3. Change directories on the Unix server to `<base_directory>/rdw12.0/log`.
   - Review the log file that was created or modified.
4. Change directories on the Unix server to `<base_directory>/rdw12.0/error`. Review the error files that were created for default_load.
5. Change directories on the Unix server to `<base_directory>/rdw12.0/install`.

6. Rerun the RDW profile and verify the MMUSER and PASSWORD variables are set to the batch user, rdw12dev and the appropriate password. At the Unix prompt, enter:

   time_static_load.ksh

   time_static_load.ksh - populates the following tables with default values:

   - `time_minute_dm`
   - `time_hour_dm`
   - `time_half_hour_dm`
   - `time_qtr_hour_dm`

7. Change directories on the Unix server to `<base_directory>/rdw12.0/log`. Review the log file that was created or modified.
8. Change directories on the Unix server to `<base_directory>/rdw12.0/error`. Review the error files that were created for time_static_load.
9. Refer to the RDW Operations Guide for more information on the log files and error files.

Populate Time Dimension

This section describes the initial load and loading of additional time into RDW. The time dimension can be loaded with a 454 calendar, 13 period time calendar or a 454 with Gregorian calendar. Populate these tables according to business requirements. If RMS is implemented, the time dimension with 454-calendar time or 454 with Gregorian calendar the calendar information can be extracted from this system. Refer to the RMS Operations Guide for details of how to extract time. After the calendar information has been extracted, proceed to section ‘Create Time in RDW’ to load it in RDW. To populate time for 454, 13 period or 454 with Gregorian calendar standalone, proceed to section ‘Create Standalone Time’. For information on the tables loaded for the Time dimension refer to the RDW Data Model documentation.

Create Standalone Time

Modify the sample text files that will serve as the data to populate the time calendar dimension with appropriate time for the environment. If a 454-time calendar or a 454-time calendar with Gregorian calendar will be used, modify the time_454*.txt. If implementing a 13 period time modify the time_13.txt file with the time calendar data. Modify the file, wkday*.txt, with the description of the first day of the fiscal week, for example ‘SUNDAY’. The entry in wkday.txt must be in all capital letters. A sample of these text files has been provided in `<base_directory>/rdw12.0/dev/install`. See the RDW Operations Guide Appendix A ‘API Flat File Specifications’ for instructions on how to modify the text files. Proceed to ‘Create Time in RDW’ to load time into the Data Warehouse.

Note: Sample text files can be used to get started in a development environment. These files need to be modified to match the fiscal calendar before creating time in a production environment

Create Time in RDW

If more time data is needed, modify the parameters below for the new time period and run the module again. To minimize the load time for adding additional time, enter only the first year to be added as the response for the first year prompt in number 5.

1. If RMS is not being used as the source of the time calendar, move on to step 2, otherwise after running time extraction in RMS (see RMS Operations Guide for details), FTP time_454.txt, start_of_half_month.txt, and wkday.txt from RMS install directory to the RDW directory `<base_directory>/rdw12.0/install.`
2. Login to the RDW database server as the UNIX user rdw12dev. Verify the RETL executable is in the path of your Unix session by typing `which rfx` at the UNIX prompt.
3. Change directories to `<base_directory>/rdw12.0/install.`
4. Verify the C compiler is in the path of your Unix session and the C compiler is in your Unix library path. At the Unix prompt, enter:

   ```
   which cc
   ```
5. Compile the module cr_time_454, cr_time_13 and cr_time_g with a standard ANSI C compiler. At the Unix prompt, enter:
   - cc –g –I. –o cr_time_454 c_utils.c cr_time_454.c
   - cc –g –I. –o cr_time_13 c_utils.c cr_time_13.c
   - cc –g –I. –o cr_time_g c_utils.c cr_time_g.c
6. Execute the `cr_time.ksh` module. At the Unix prompt enter:
   - `cr_time.ksh`
7. This script will prompt for the calendar type. Choose 1 for 454 time, 2 for 13 period time and 3 for 454 with Gregorian time.
8. At the prompt enter the 4-digit year for the beginning and ending of the time calendar:

   ```
   Please enter first year to be loaded:
   Please enter last year to be loaded:
   ```

   Note: To determine the beginning and ending fiscal year, refer to the text file modified above. Verify all months or periods are included in the text file for the first year; no partial years are allowed.

   - One text file will be generated in the install directory for each dimension table.
9. At the UNIX prompt, for 454 time calendar or 13 period time calendar enter:
   - time_load.ksh
   - time_trnsfrm_load.ksh
10. At the UNIX prompt, for 454 time with Gregorian time calendar enter
    - time_load.ksh
    - g_time_load.ksh
    - time_trnsfrm_load.ksh
    - g_time_trnsfrm_load.ksh

11. Change directories on the Unix server to <base_directory>/rdw12.0/log. Review the log file that was created or modified.
12. Change directories on the Unix server to <base_directory>/rdw12.0/error. Review the error files that were created for time_load and time_trnsfrm_load. Review also the error files that were created for g_time_load and g_time_trnsfrm_load if 454 time with Gregorian calendar is used.
13. Refer to the RDW Operations Guide for more information on the log files and error files.

> **Note:** DAY_IDNT serves as a surrogate key for day of the time. It can represent 454 time, 13 period time or Gregorian time. Since it is a surrogate key, it does not have meaning to it as long as it is unique across all the days of the time used in RDW.

Populate Static Dimension Data

There are several dimension tables in RDW that must be loaded before loading Voucher Facts and executing some MicroStrategy Reports. In order to determine the information needed for these tables, refer to the RDW Operations Guide, 'API Flat File Specifications'. Sample flat files have been provided in the `<base_directory>/rdw12.0/install directory,` these tables must be populated according to business requirements.

The tables to be populated are time_last_yr_by_day_lfl_dm, time_last_yr_by_wk_lfl_dm, vchr_age_band_dm. See the RDW Data Model documentation for more information on the tables and the column descriptions to be loaded.

1. Login to the RDW database server as the UNIX user rdw12dev. Verify the RETL executable is in the path of your Unix session by typing:
   - which rfx
2. Change directories on the Unix server to `<base_directory>/rdw12.0/install`. This directory contains the code for all default data install scripts.
3. The script timelfldm.ksh loads the following Time 'Like for Like' tables that are used exclusively by the MicroStrategy reports:
   - time_last_yr_by_day_lfl_dm
   - time_last_yr_by_wk_lfl_dm
4. Load the Time 'Like for Like' tables by executing the script. At the Unix prompt, enter:

   ```
   timelfldm.ksh
   ```
5. Change directories on the Unix server to `<base_directory>/rdw12.0/log`. Review the log file that was created or modified.
6. Change directories on the Unix server to `<base_directory>/rdw12.0/error`. Review the error files that were created.
7. Change directories on the Unix server to `<base_directory>/rdw12.0/install`. The script, vchragedm.ksh, loads the table, vchr_age_band_dm, which is used by the voucher fact load batch module. At the Unix prompt, enter:

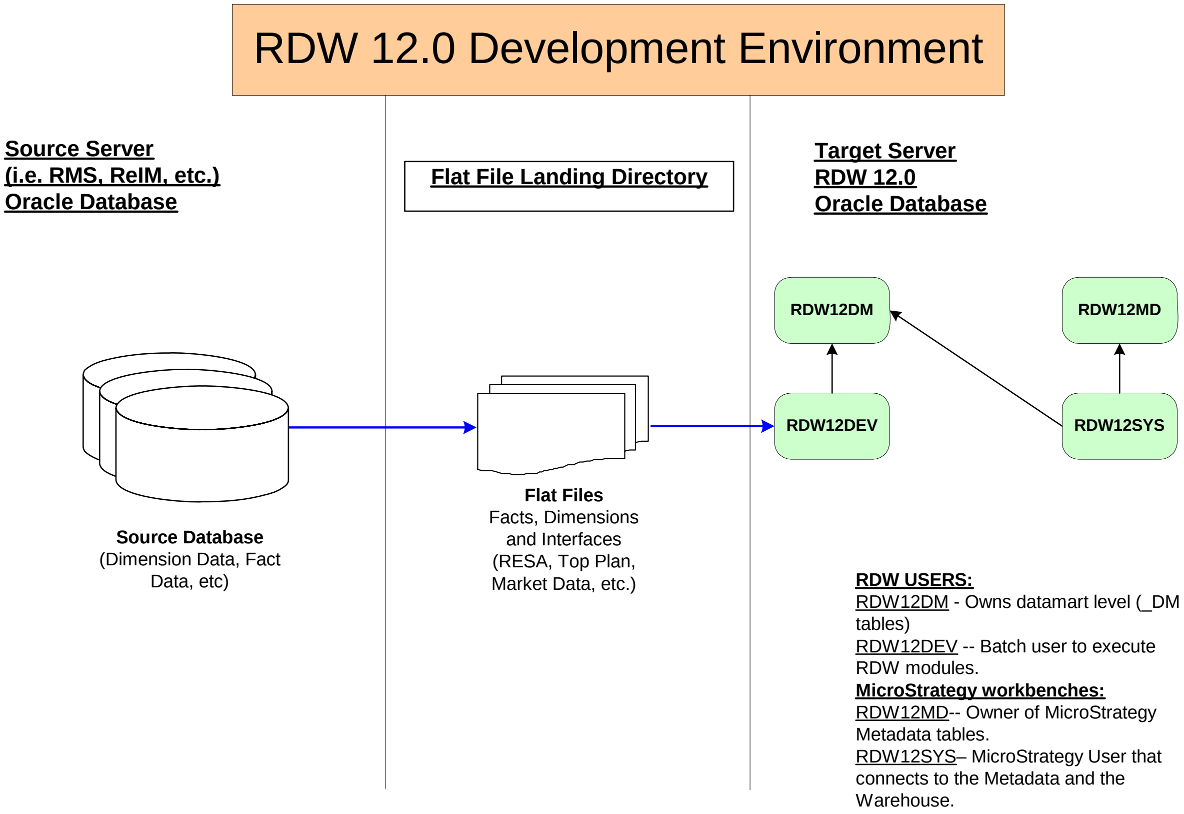
   ```
   vchragedm.ksh
   ```

8. Change directories on the Unix server to `<base_directory>/rdw12.0/log`. Review the log file that was created or modified.
9. Change directories on the Unix server to `<base_directory>/rdw12.0/error`. Review the error files that were created.
10. Refer to the RDW Operations Guide for more information on the log files and error files.

A

Appendix: Development Database Schema Implementation

The following figure shows the source system and RDW development schemas that are referred to in this installation guide.


RDW 12.0 Development Environment
Source Server
(i.e. RMS, ReIM, etc.)
Oracle Database
Flat File Landing Directory
Target Server
RDW 12.0
Oracle Database
RDW12DM
RDW12MD
RDW12DEV
RDW12SYS
Source Database
(Dimension Data, Fact Data, etc)
Flat Files
Facts, Dimensions and Interfaces (RESA, Top Plan, Market Data, etc.)

RDW USERS:
RDW12DM - Owns datamart level (_DM tables)
RDW12DEV -- Batch user to execute RDW modules.
MicroStrategy workbenches:
RDW12MD-- Owner of MicroStrategy Metadata tables.
RDW12SYS– MicroStrategy User that connects to the Metadata and the Warehouse.

B

Appendix: RDW Code Tree

The following table describes the contents of each of the RDW code tree directories created during the RDW installation.

Path	Directory	Description
<base_directory>	dbasql	This directory contains all SQL scripts necessary to maintain the permissions for the database users.
<base_directory>	batch	Empty directory used for development and testing purposes only.
<base_directory>	data	This directory contains the text files that serve as the input to RDW RETL load modules. For clients with Retek applications such as RMS, ReIM, etc, these text files can be generated in these source applications. The text files should be FTP from the source application to the RDW data directory. Directory is empty on installation.
<base_directory>	error	This directory holds all program error files, and status files. Directory is empty on installation.
<base_directory>	install	This directory contains all RETL modules and SQL scripts needed only at installation.
<base_directory>	log	This directory holds log files of program execution. Directory is empty on installation.
<base_directory>	rfx	This directory contains subdirectories for all the code and files related directly to RETL
<base_directory>/rfx	bookmark	This directory contains a file created during execution of each module to track the execution of the module. Files are deleted upon successful completion of module. Directory is empty on installation.
<base_directory>/rfx	etc	This directory contains files that hold variables used by RDW batch modules. The configuration file is found in this directory.
<base_directory>/rfx	include	This directory contains files that hold string language translations used by RDW batch modules.
<base_directory>/rfx	lib	This directory contains all RDW library code
<base_directory>/rfx	schema	This directory contains all RDW schema files used with each module.
<base_directory>/rfx	src	This directory contains RDW source code.

C

Appendix: Oracle 10g Database Creation Scripts

```
#########################################################################
######
# Oracle 10.2.0.x Parameter file
## NOTES: Before using this script:
#        1.  Change <datafile_path>, <admin_path>, <utl_file_path>, and
<hostname>
#            values as appropriate.
#        2.  Replace the word SID with the database name.
#        3.  Size parameters as necessary for development, test, and
production environments.
# -----------------------------------------------------------------------
-
# MAINTENANCE LOG
#
# Date     By          Parameter           Old/New         Notes
# +------+ +---------+ +-----------------+ +-------------+ +-------------
+
# 02/20/06 Oracle       NA                  NA              creation
#
#########################################################################
######
# -----------------------------------------------------------------------
# The following SGA parameters are CRITICAL to the performance of the
# database. The following settings are based on 1GB of allotted memory.
# The SGA is composed of:
#  db_cache_size, log_buffer,java_pool_size, large_pool_size,
#  shared_pool_size
# -----------------------------------------------------------------------
db_cache_size                   = 256M
java_pool_size                  = 150M        # 150M for initial db
creation
log_buffer                      = 10485760
shared_pool_size                = 350M        # 350M for initial db
creation
shared_pool_reserved_size       = 35M         # 10% of shared_pool_size
# -----------------------------------------------------------------------
---------
# The following parameters do not affect SGA size;
# -----------------------------------------------------------------------
---------
audit_file_dest                                = <admin_path>/adump
background_dump_dest                    = <admin_path>/bdump
compatible                              = 10.2.0
control_files                 = (<datafile_path>/control01.ctl
                                ,<datafile_path>/control02.ctl)
core_dump_dest                                 = <admin_path>/cdump
db_block_size                                  = 8192 # Default is 2k;
adjust before db creation, cannot change after db is created
db_file_multiblock_read_count    = 16 # Platform specific (max io
size)/(block size)
db_name                                              = SID
job_queue_processes                     = 5 # Oracle Retail required;
number of cpu's + 1
local_listener =  "(ADDRESS=(PROTOCOL=TCP)(HOST=<hostname>)(PORT=1521))"
nls_calendar                  = GREGORIAN
nls_date_format                                = DD-MON-RR # Oracle
Retail required; if RDW database see later entry for proper format
```

```
nls_language                            = AMERICAN  # Default
nls_numeric_characters                  = ".,"      # Should be
explicitly set to ensure all users/batch get the same results
nls_sort                                       = BINARY    # Should be
explicitly set to ensure all sessions get the same order
nls_territory                           = AMERICA   # Default
open_cursors                            = 900       # Oracle Retail
required (minimum=900); default is 50
optimizer_features_enable        = 10.2.0.1
optimizer_mode                                 = CHOOSE    # Oracle
Retail required
pga_aggregate_target          = 100M
plsql_optimize_level          = 2         # 10g change; use this setting
to optimize plsql performance
plsql_debug                   = false     # 10g change; use this setting
to optimize plsql peformance
processes                                      = 500               # Max
number of OS  processes that can connect to the db
query_rewrite_enabled                   = TRUE      # Oracle Retail
required for function-based indexes
session_cached_cursors        = 900       # Oracle Retail required; 10g
uses to cache sql cursors in pl/sql
undo_management                         = AUTO
undo_retention                                 = 1800      # Currently
set for 30 minutes; set to avg length of transactions in sec
undo_tablespace                         = undo_ts
user_dump_dest                                 = <admin_path>/udump
utl_file_dir                  = <utl_file_path>
workarea_size_policy          = auto        # Should be set to auto when
pga_aggregate_target is set

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

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

-------------------------------------------------------------------------
-- Script:      crdb1.sql
--- Execute as:  sysdba
--- Note:        Before running this script:
---                 Modify <datafile_path> values.
---                 Modify SID values.
---                 Adjust sizes for redo logs, datafiles and tempfile
-------------------------------------------------------------------------
```

```
spool crdb1.log

STARTUP NOMOUNT pfile=${ORACLE_HOME}/dbs/initSID.ora
CREATE DATABASE "SID"
        MAXDATAFILES 1000
        CHARACTER SET UTF8
    DATAFILE
        '<datafile_path>/system01.dbf' SIZE 500M AUTOEXTEND ON NEXT 100M
MAXSIZE 2000M
    LOGFILE
      GROUP 1 ('<datafile_path>/redo1a.log') SIZE 1000M,
      GROUP 2 ('<datafile_path>/redo2a.log') SIZE 1000M,
      GROUP 3 ('<datafile_path>/redo3a.log') SIZE 1000M
DEFAULT TEMPORARY TABLESPACE temp TEMPFILE '<datafile_path>/temp01.dbf'
SIZE 5000M EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M
UNDO TABLESPACE undo_ts DATAFILE '<datafile_path>/undo_ts01.dbf' SIZE
5000M
SYSAUX DATAFILE '<datafile_path>/sysaux01.dbf' SIZE 500M AUTOEXTEND ON
NEXT 100M MAXSIZE 2000M
;

exit
spool off

---------------------------------------------------------------------
--- Script:         crdb2.sql
--- Execute as:     sysdba in 10.1.0.2 databases or higher
--- Note:           This script installs the data dictionary views in
---  addition to granting necessary privileges to public.
---------------------------------------------------------------------

spool crdb2.log
REM # install data dictionary views:
PROMPT Running catalog.sql
@$ORACLE_HOME/rdbms/admin/catalog.sql;
PROMPT Running catblock.sql
@$ORACLE_HOME/rdbms/admin/catblock.sql;
PROMPT Running catproc.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql;
PROMPT Running catoctk.sql
@$ORACLE_HOME/rdbms/admin/catoctk.sql;
PROMPT Running catrep.sql
@$ORACLE_HOME/rdbms/admin/catrep.sql;
PROMPT Running owminst.plb
@$ORACLE_HOME/rdbms/admin/owminst.plb;

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


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

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

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

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

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

spool off
exit

-----------------------------------------------------------------
--- Script:        crdb3.sql
--- Execute as:    sysdba in 10.1.0.2 databases or higher
--- Note:          This script installs java and xml components;
---                Do not change the order of the statements below due to
--                 dependencies
-----------------------------------------------------------------


spool JServer.log
@$ORACLE_HOME/javavm/install/initjvm.sql;
@$ORACLE_HOME/xdk/admin/initxml.sql;
@$ORACLE_HOME/xdk/admin/xmlja.sql;
@$ORACLE_HOME/rdbms/admin/catjava.sql;
@$ORACLE_HOME/rdbms/admin/catexf.sql;
spool off

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

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

@$ORACLE_HOME/rdbms/admin/utlrp.sql
```

D

Appendix: Tablespace Creation Scripts

```
-------------------------------------------------------------------------
------------
--- Script:        create_rdw_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_rdw_tablespaces.log

CREATE TABLESPACE cache_data
   DATAFILE '<datafile_path>/cache_data01.dbf' SIZE 15M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE dm_dim_data
   DATAFILE '<datafile_path>/dm_dim_data01.dbf' SIZE 400M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE dm_dim_index
   DATAFILE '<datafile_path>/dm_dim_index01.dbf' SIZE 300M
 AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE dm_fact_data
   DATAFILE '<datafile_path>/dm_fact_data01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE dm_fact_index
   DATAFILE '<datafile_path>/dm_fact_index01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE dim_mtx_data
   DATAFILE '<datafile_path>/dim_mtx_data01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE dim_mtx_index
   DATAFILE '<datafile_path>/dim_mtx_index01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE ms_meta_data
   DATAFILE '<datafile_path>/ms_meta_data01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE user_data
   DATAFILE '<datafile_path>/user_data01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 50M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
spool off
exit
```

```
-------------------------------------------------------------------------
--- Script:        create_rdw_dwi_tablespace.sql
--- Execute as:    sysdba
--- Note:          Before running this script:
---                  Run this script in the RMS database if it is
separate from the RDW database.
---                  Modify <datafile_path> values.
---                  Modify datafile storage parameters and sizes based
on partitioning strategy.
-------------------------------------------------------------------------
spool create_rdw_dwi_tablespace.log

CREATE TABLESPACE user_data
   DATAFILE '<datafile_path>/user_data01.dbf' SIZE 300M
    AUTOEXTEND ON NEXT 50M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO

;
```