

Retek® Data Warehouse 10.0



Database Installation Guide



The software described in this documentation is furnished under a license agreement 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.

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.

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
888.61.RETEK (toll free US)
+1 612 587 5000

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

Retek[®] Data Warehouse[™] 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This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws.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:

©2002 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 e-mail, 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.

Contact Method Contact Information

Internet (ROCS) www.retek.com/support
Retek's secure client Web site to update and view issues

E-mail support@rettek.com

Phone US & Canada: 1-800-61-RETEK (1-800-617-3835)
World: +1 612-587-5800
EMEA: 011 44 1223 703 444
Asia Pacific: 61 425 792 927

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 – Introduction.....	1
Typographic conventions	1
Command syntax	1
Chapter 2 – Data Warehouse server and system requirements	3
Data warehouse server (target server)	3
Hardware requirements	4
Software requirements and configuration matrix	5
Capacity planning	6
Client system requirements	6
Chapter 3 – Data Warehouse interface Unix environment setup.....	7
Installation instructions	7
Create Unix user accounts	7
Create Unix directories	7
Alter directory ownership and privileges (optional)	10
Copy profiles for Retek users	10
Chapter 4 – RMS/DWI Oracle database configuration	11
General installation notes	11
Installation instructions	11
Chapter 5 – RDW Unix environment setup	17
Installation instructions	17
Create Unix user accounts	17
Create Unix directories	18
Alter directory ownership and privileges (optional)	19
Copy profiles for Retek users	20

Chapter 6 – RDW RDBMS installation and configuration on Oracle	21
General installation notes	21
Oracle Concepts for the Retek Data Warehouse	21
Installation instructions	21
Oracle database configuration	21
Chapter 7 –RDBMS installation and configuration on DB227	
General installation notes	27
DB2 Concepts for the Retek Data Warehouse	27
Installation instructions	29
DB2 database configuration	29
Chapter 8 – RDBMS installation and configuration on Teradata	33
General installation notes	33
Teradata Concepts for the Retek Data Warehouse.....	33
Installation instructions	34
Teradata database configuration	34
Chapter 9 – Default data population	37
ETL installation and configuration	37
Data population	37
Populate datamart default data	37
Populate time dimension	38
Populate static dimension data	40
Appendix A – Oracle information	43
Compile scripts for DWI Pro*C modules	43
Sample Oracle Net Services Files for the Server	44
listener.ora	45
tnsnames.ora	46

Appendix B – DB2 information	47
Sample database configuration parameters on DB2.....	47
Appendix C – Additional information	49
Development Database Schema implementation	49

Chapter 1 – Introduction

Typographic conventions

monospace

Monospace type indicates UNIX commands, directory names, usernames, pathnames, and filenames.

brackets []

Words enclosed in brackets indicate key names (for example, Press [Return]).

Note: Brackets have a different meaning when used in command syntax.

italics

Italic type indicates a variable, including variable portions of filenames. It is also used for emphasis.

i

UPPERCASE

Uppercase letters indicate Structured Query Language (SQL) reserved words, initialization parameters, and environment variables.

Command syntax

UNIX command syntax appears in monospace font and assumes the use of the Korn shell. The ">" character at the beginning of UNIX command examples should not be entered at the prompt.

**backslash ** A backslash indicates a command that is too long to fit on a single line. Enter the line as printed (with a backslash) or enter it as a single line without a backslash:

```
imp system/password log=rdw10_md.log \
file=rdw10md.dmp ignore=y grants=n \
fromuser=rdw10md touser=rdw10md
```

In the following instructions, the character ‘\’ in a path name indicates a DOS-based file structure, for example the CD. The character ‘/’ in a path name indicates a Unix-based file structure.

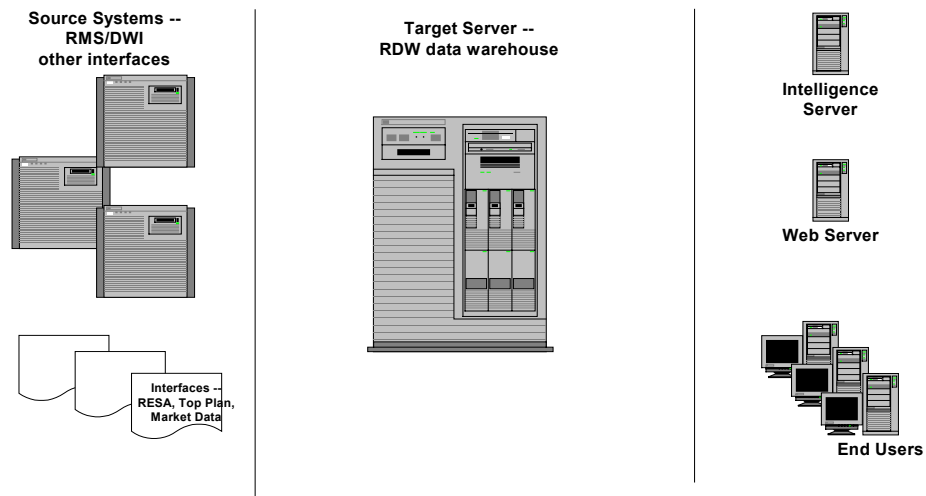
Chapter 2 – Data Warehouse server and system requirements

Data warehouse server (target server)

Retek Data Warehouse (RDW) version 10.0 architecture supports a database independent platform. RDW supports three database platforms: Oracle, DB2 or Teradata. This release of RDW incorporates an optional interface with Retek Merchandising System (RMS) 10.0, which can be a source for the data warehouse. This interface has been named the Data Warehouse Interface (DWI), which is a part of the RMS database and is located on the source server. The data warehouse can also operate as a standalone product and be fed from other legacy systems. If you are using RMS, follow the guidelines set forth in the RMS documentation regarding the server requirements for the source server. The data warehouse server or RDW server is considered the target server. It is recommended that the source system is on a separate server from the data warehouse.

RDW Architecture on Oracle and DB2

(see chapter 6 for Teradata architecture)



These are the hardware and software requirements necessary to install and operate the Retek Data Warehouse (RDW) software.

General requirements for a data warehouse server capable of running RDW include:

- Unix based OS
- TCP/IP Network
- Standard ANSI C compiler

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)

Note: This reflects the amount of space needed to install the database 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 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. Retek can assist with making these determinations by providing information on database size estimates, server architecture, and so on.

Software requirements and configuration matrix

The RDW 10.0 Configuration Matrix shows the currently supported configurations at the time of release. If a required configuration that is not supported, verify with Retek Customer Support to see if the configuration is now supported.

Database*	Hardware Platform /Operating System (Target Database Server)**	Hardware Platform /Operating System (Source Database Server)	Load Server	ETL tool (binary compiled files only for each hardware platform and operating system for database specified)
IBM DB2 Universal Database - Extended Enterprise Edition 7.2 (32-bit)	AIX 4.3.3	Optional interface with RMS 10.0, which includes the Data Warehouse Interface (DWI) environment, use guidelines recommended by RMS documentation. If standalone source, no server requirements but must follow API guidelines.		RIB ETL 1.7 for DB2 on supported hardware platform.
NCR Teradata V2R4.0 or R4.1	WorldMark 5xxx series – UNIX WorldMark 4xxx series – UNIX S2x series – UNIX	Optional interface with RMS 10.0, which includes the Data Warehouse Interface (DWI) environment, use guidelines recommended by RMS documentation. If standalone source, no server requirements but must follow API guidelines.	Sun Solaris 8 HP-UX 11i(***) AIX 4.3.3	RIB ETL 1.7 for Teradata on supported hardware platform. The RIB ETL application will be on the load server along with the Teradata load utilities.
Oracle RDBMS Enterprise Edition with Partitioning Option 9.0.1.2 (64-bit)	Sun Solaris 8 HP-UX 11i AIX 4.3.3	Optional interface with RMS 10.0, which includes the Data Warehouse Interface (DWI) environment, use guidelines recommended by RMS documentation. If standalone source, no server requirements but must follow API guidelines.		RIB ETL 1.7 for Oracle on supported hardware platform.

Notes:

*Review the database documentation for minimum guidelines and requirements.

** Other configurations may exist call Retek Customer Support for currently available configurations. Future availability will depend on demand. Testing required prior to support.

*** HP-UX 11i is available upon request.

Other software requirements for the data warehouse server include:

- MicroStrategy 7
- RDW Web 2.x
- Oracle Pro*C Precompiler 9.0.1 (on source server only)

Capacity planning

If you have not yet estimated the minimum disk space for your Retek Data Warehouse, contact Retek Services, services@retek.com, to request a RDW Database Sizing Questionnaire kit. When you have completed the Questionnaire return it to Retek Services for an estimate on the minimum disk space required for your Retek Data Warehouse. Refer to your RDW sizing information to determine tablespace, table, index, and sizes as appropriate for planning your production environment. The database size created by the install scripts in this document is based on a typical development environment.

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 MicroStrategy Middle Tier Installation Guide and the RDW Web Installation Guide for other client requirements.
- X-term station or application (such as Exceed) for Oracle installation. Oracle 9.0.x requires an X-term for GUI installation.

Chapter 3 – Data Warehouse interface Unix environment setup

Installation instructions

The Data Warehouse Interface (DWI) is a module released with RDW 10.0 that allows the Retek Merchandising System (RMS) 10.0 to interface with this release of the RDW. The DWI must be installed on the same server and in the same database as RMS.

If you are not using RMS, do not install this module, but continue on to Chapter 5, Retek Data Warehouse Unix Environment setup. These instructions assume that RMS 10.0 is installed and running. They also assume that an Optimal Flexible Architecture (OFA) compliant file structure has been created on the Unix server.

Create Unix user accounts

- 1 Log in as the `root` user.
- 2 Create Unix groups for the following environments:
 - `dba` - admin group which controls DWI access
 - `dev` - development group
- 3 Create the following Unix users, using `ksh` as the default shell:
 - `retex` - `dba` group
 - `dwidew` - `dev` group

Note: These steps must be completed exactly as described, since these Unix accounts are referenced later in installation scripts. To successfully run the DWI batch modules, environmental variables are needed and are set in the profiles of these Unix user accounts. These profiles are set in a later step.

Create Unix directories

Create Unix directories for the DWI database directory structure and the code directory structure.

Note: Database directories are used to provide scripts used in the creation of tablespaces and users. The code directory structure stores source code, error and log directories.

- 1 Create the DWI database directory structure:
 - a Log in to the Unix server as `oracle`. `Oracle` will own the database directory structure.
 - b FTP (in binary mode) the tar file, `RDW_10_CD\Install\dw10_db.tar`, to `$ORACLE_BASE/admin` on the Unix server where the RMS database is located. The `$ORACLE_BASE` directory is a working directory that should be designated at this time.
 - c Extract the tar file to create DWI database admin directories. As the `oracle` user, extract the `dw10_db.tar` file by typing:


```
>tar -xvf dw10_db.tar
```

Note: A duplicate of this file structure including all subsequent folders and files can be found on the CD-ROM in the directory

RDW_10_CD\Install\dw10.0\database.

Once expanded, the directory structure looks as follows:

```
$ORACLE_BASE/admin/dw10/create  
/ddl
```

2 Create the DWI code directory structure:

- a Log in to the Unix server as `retex`. Retek will own the code directory structure.
- b FTP (in binary mode) the file `RDW_10_CD\Install\dw10_code.tar` to a base directory, the permanent location for the DWI code directory, on the Unix server where the RMS database is located. Designate a working directory, the base directory, at this time.
- c Extract the tar file to create DWI code directory. As the `retex` user, extract the `dw10.tar` file in the base directory by typing:

```
>tar -xvf dw10_code.tar
```

Note: A duplicate of this file structure including all subsequent folders and files can be found on the CD-ROM in the directory `RDW_10_CD\Install\dw10.0\unix`.

Once expanded, the directory structure looks as follows:

```
<base_directory>/dwi10.0/dbasql/
    /dev/batch
        /data
        /error
        /install
        /log
        /oracle/lib/bin
            /lst
            /src
        /proc/bin
            /src
            /lst
        /sqlldr/data/thread_1
    /sqlplus
    /rfx/bin
    /etc
    /lib
    /schema
    /src
    /retek/bin
    /sample_profiles
```

Note: We highly recommend that clients have a separate server that is for production only. When the time comes for a production environment, create a prd directory, then copy the dev directory structure to the production server and use the <base_directory>/dwi10.0/prd as the code directory structure on that server. A prd group is needed and a separate prd user should be created as well.

The <thread_nn> directories are the thread directories used by multi-threaded Pro*C batch programs. All DWI batch programs that write to flat files have been set up to write to a common data directory referenced by the environment variable \$atldrdata. This points by default to the <base_directory>/dwi10.0/dev/oracle/sqlldr/data directory but can be changed when the dwi_profile is set up for the dwidev Unix account in a later step.

Consequently, during multi-threaded usage, the device can become I/O bound very quickly. The multi-threaded DWI batch programs have therefore been altered to write to the directory `$atldrdata/thread_nn`, where `nn` is the actual thread number. Because each thread now writes to its own directory, the I/O load can be balanced across devices. To do this, set up the appropriate number of `thread_nn` (for example, `thread_1` to `thread_12`) directories in the `$atldrdata` directory as symbolic links to other devices using the Unix command

```
ln -s <device path>/thread_nn <atldrdata path>/thread_nn. At least one thread_nn directory must exist. thread_1 is provided by default, since all DWI Pro*C batch programs must be run with at least one individual thread.
```

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>/dwi10.0` directory.
- 2 Change the following as indicated in the table below:
 - the ownership (`chown -R <owner> <directory>`)
 - the group (`chgrp -R <user group> <directory>`)
 - and the privileges (`chmod -R <privs> <directory>`)

Directory	Owner	Group	Privilege
rettek	rettek	Dbal	775
dev	dwidev	Dev	775
dbasql	oracle	Dbal	774

For example: `cd <base_directory>/dwi10.0`
`chown -R dwidev dev`
`chgrp -R dev dev`
`chmod -R 775 dev, etc...`

Copy profiles for Retek users

For each of the Retek users, copy the profiles.

- 1 At the Unix prompt, enter
`cd <base_directory>/dwi10.0/rettek/sample_profiles.`

In this directory is a sample of the profile that needs to be set up for each Unix user account.

- 2 Copy the profile to the home directory of each user. For example, `dwi_profile` should be copied to `.profile` in the home directory of the `dwidev` user that was created.

Note: The `dwi_profile` by default is set up for a development environment. Change the indicated variables as necessary for other environments, such as `dwitst` or `dwiprd`. Any variable that may require modification is 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.

Chapter 4 – RMS/DWI Oracle database configuration

General installation notes

We recommend the following instructions for the development and test environments only. Consult your RDW sizing estimate to determine tablespace, table, index, and other sizes as appropriate for planning your production environment. When planning your production or test environments, modify the scripts and instructions as necessary for, tablespace requirements, and table and index storage parameters, and so on.

Installation instructions

The Data Warehouse Interface (DWI) is a module released with RDW 10.0 that allows the Retek Merchandising System (RMS) 10.0 to interface with this release of the RDW. The DWI must be installed on the same server and in the same database as RMS.

If you are not using RMS, do not install this module, but continue on to Chapter 5, Retek Data Warehouse Unix Environment setup. These instructions assume that RMS 10.0 is installed and running. They also assume that an Optimal Flexible Architecture (OFA) compliant file structure has been created on the Unix server.

Note: Clients should become familiar with the DWI development environment before setting up a production system.

Create database objects (tablespaces and users)

The scripts to create the DWI users and tablespaces are in Unix subdirectories of the **\$ORACLE_BASE/admin/dwi10/create** directory, which were created in an earlier step.

1 Create Tablespace:

The Unix directory **\$ORACLE_BASE/admin/dwi10/create** contains scripts for creating the three tablespaces needed by the DWI application.

Note: We assume that a temporary tablespace as well as undo segment and RMS data/index tablespaces already exist.

- a Log in to Unix as Oracle.
- b Modify the script, `crtbspc_dwi.sql` to include the correct Unix files system paths for the datafiles, as well as proper sizing/storage parameters for the environment being set up. The default sizing of the tablespaces in these create scripts assumes they are being used to set up a development environment.
- c The script assumes the RMS database name is `rms10`, so change the file structure and datafile names to reflect the actual RMS database name if desired.

- d Alter the paths to fit appropriately into the existing RMS datafile structure.
- e Change directories to \$ORACLE_BASE/admin/dwi10/create
- f Log in to SQL*Plus as system and run the following script:
SQL>@crtbspc_dwi.sql

2 Create Users:

- a Modify the script crusers_dwi.sql as needed prior to running. This script can then be used to create the default DWI Oracle users and grant the appropriate system privileges. Refer to the Development Database Schema implementation section in Appendix C for descriptions of each user. Modify the script by changing the passwords.
- b Log in to SQL*PLUS as system and execute:
SQL>@crusers_dwi.sql
- c To secure the passwords in this script, change the privileges on the script by entering:
>chmod 700 crusers_dwi.sql

Create DWI extraction tables

- 1 Change directories to \$ORACLE_BASE/admin/dwi10/ddl.
- 2 Log into SQL*Plus as dwi10ex and execute the following script:
SQL>@dwi10ex_ora.sql.

This creates all of the interface tables. This script calls subsequent scripts and has been generated to execute all scripts in the same directory.

Because some of the views depend on PL/SQL objects and synonyms to RMS tables created in a later step, they will not be successfully created in this step. The invalid objects will be compiled in a later step.

Create and compile DWI PL/SQL

- 1 Change directories on the Unix server to
<base_directory>/dwi10.0/dev/sqlplus.

This directory contains the code for all DWI PL/SQL objects (including packages and procedures).
- 2 Edit the script rdwtrigb.pls, changing the parameter L_file_path to suit your environment. This is a path to the util_file directory specified in the RMS database parameter file.
- 3 Connect to SQL*Plus as dwi10ex and run the following script:
SQL> @dwi_plsqlload.sql.

This script subsequently calls all other DWI PL/SQL executables in the directory and then creates the DWI packages and procedures.

Set up DWI user permissions

The following scripts set up all necessary synonyms and grants for the DWI environment, as well as grant the appropriate security level for the `dwi10dev` batch user to access the RMS tables during a batch run.

- Refer to the header of each file for information on which user should execute the script and more detailed information on what the script does.
- The `dwi_users.sql` script has been set up to reference the default DWI usernames, so verify that they are correct and change to suit your environment if necessary.
- The scripts will spool to the `/tmp` directory, which should already exist.
- When prompted by any of the scripts for a table/view name, press **[Enter]** to create permissions for all objects in the schema.

- 1 Change directories on the Unix server to
`<base_directory>/dwi10.0/dbasql`
- 2 Connect to SQL*Plus as `rmsdev100` (the RMS schema owner) and execute the following scripts:

```
SQL>@gs_rms10.sql
```

```
SQL>@g_security.sql
```

- 3 Verify the scripts executed correctly by reviewing the following files:

```
g_rms100_1.lst, s_rms100_1.lst.
```

- 4 Connect to SQL*Plus as `dwi10ex` and execute the following scripts:

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

```
SQL>@gs_dwi10_ex.sql
```

- 5 Verify the scripts executed correctly by reviewing the following files:

```
g_dwi100_ex1.lst, s_dwi100_ex1.lst
```

Note: The `invalids` script may need to be run several times to validate all of the interrelated objects.

- 6 Connect to SQL*Plus as `sys` and execute the following script:

```
SQL>@g_public_privs.sql
```

Note: These scripts are also used for maintenance. If you need to drop or add a table owned by `dwi10ex` or `rmsdev100` (the RMS Schema owner), you need to 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 `dwi_users.sql` script to alter the usernames for the environment you are creating.

Create and compile DWI triggers

- 1 Change directories on the Unix server to
`<base_directory>/dwi10.0/dev/sqlplus`. This directory contains the code for all DWI triggers.
- 2 Connect to SQL*Plus as `dwi10ex` and execute the following script:
`SQL> @dwi_trigload.sql`
 This script subsequently calls all other DWI executables in the directory and creates the DWI triggers.

Populate DWI restart and recovery default values

Perform this procedure to populate the restart and recovery data for the DWI modules.

- 1 Change directories on the Unix server to
`<base_directory>/dwi10.0/dev/install`.
- 2 Log in to SQL*Plus as `dwi10dev` and execute the following script:
`SQL>@dwi_rr_load.sql`
 This script sets up the DWI batch modules to run with one thread and an array size of 1000 records. Refer to the RDW Program Overview chapter in the *Retek Data Warehouse Operations Guide* for details on how to set up the system restart/recovery elements to suit your needs, including using multi-threading and larger array sizes.
 These parameters are environment specific, so alter them as appropriate based on your data and hardware configuration. In particular, the array size (`commit_max_ctr`) should almost always be greater than 1.
 The script
`<base_directory>/dwi10.0/dev/oracle/proc/bin/dwi_rr_refresh.sql` can be used as a template for a daily restart/recovery update process, but does not need to be run to set up the environment.

Configure RIB ETL

- 1 Log in to the Unix server with the `dwidev` account.
- 2 Change directories to `<base_directory>/dwi10.0/dev/rfx/etc`.
- 3 Modify the `config.env` script:
 - a Change the `DBNAME` variable to the name of the RMS database.
 - b Change the `RMS_OWNER` variable to the username of the RMS schema owner.

Compile DWI batch libraries

- 1 Log in to the Unix server with the `dwidev` account.

Note: If you are setting up a test or production environment, log in with the appropriate Unix account -- `dwitst` for the test system, `dwiprd` for the production system.

If the account is set up with the proper profile information in the `.profile`, the environment variables will be set up by logging into the appropriate Unix user. The key environment variables for compiling batch libraries and programs are: `$ORACLE_HOME`, `$MMUSER`, `$PASSWORD`, and `$RETEK_HOME`.

See the *Compile scripts* section in Appendix A for more information. The only compile script that may require manual modification is `setplatflags.ksh` located in `<base_directory>/dwi10.0/retek/bin`, which contains platform-specific compile flags.

- 2 Change directories to
`<base_directory>/dwi10.0/dev/oracle/lib/src`.
- 3 At the Unix prompt, enter:
`>rdwlcomp dwi100 |tee dwilib.log`
- 4 Verify there were no errors in the `dwilib.log`.

Compile DWI batch programs

- 1 Log in as the `dwidev` Unix user.
- 2 Change directories to
`<base_directory>/dwi10.0/dev/oracle/proc/src`.
- 3 At the Unix prompt, enter:
`>rdwfcall -f dwiproc.log`
- 4 Verify there were no errors in the `dwiproc.log`.

Chapter 5 – RDW Unix environment setup

Installation instructions

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:
 - `retek` - `dba` group
 - `rdw10dev` - `dev` group

The following users are only needed for DB2 installations.

- `rdw10dm` – `dev` group (DB2 only)
- `rdw10md` – `dev` group (DB2 only)
- `rdw10sys` – `dev` group (DB2 only)

Note: These steps must be completed exactly as shown, since these user accounts are referenced later in installation scripts. To successfully run the RDW modules, environment variables are needed that are set in the profiles of these Unix user accounts. These profiles are set in a later step.

- 4 Identify the user that will own the database software. This user will be referred to as *rdwdba*.
 - `DB2` – DB2 instance owner
 - `Oracle` – Oracle software and database owner
 - `Teradata` – Teradata database owner

Create Unix directories

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.

1 Create RDW database directory structure:

- a Log in to the Unix server as *rdwdba* account, identified in the step above. This user will own the database directory structure.
- b FTP (in binary mode) the tar file, RDW_10_CD\Install\rdw10_db.tar, to <database_directory>/admin on the Unix server where the rdw10 database will be created. The <database_directory> directory is a working directory that you should designate at this time.
- c Extract the tar file to create RDW database admin directories in the <database_directory>/admin. As the *rdwdba* user, extract the rdw10_db.tar file by typing:

```
>tar -xvf rdw10_db.tar
```

Note: A duplicate of this file structure including all subsequent folders and files can be found on the CD-ROM in the directory RDW_10_CD\Install\rdw10.0\database.

Once expanded, the directory structure looks as follows:

```
<database_directory>/admin/rdw10/create
                                /ddl/md/ora
                                /db2
                                /ter
                                /dm
```

2 Create RDW code directory structure:

- a Log in to the Unix server as *retex*. Retek will own the code directory structure.
- b FTP (in binary mode) the file RDW_10_CD\Install\rdw10_code.tar 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 10.0 database will be created. The base directory is a working directory that you should designate at this time.
- c Extract the tar file to create the RDW code directory. As the *retex* user, extract the rdw10_code.tar file in the base directory specified above by entering:

```
>tar -xvf rdw10_code.tar
```

Note: A duplicate of this file structure, including all subsequent folders and files, is on the CD-ROM in the directory RDW_10_CD\install\rdw10.0\unix.

Once expanded, the directory structure looks as follows:

```
<base_directory>/rdw10.0/dbasql
                        /dev/batch
                        /data
                        /error
                        /install
                        /log
                        /rfx/bin
                        /etc
                        /lib
                        /schema
                        /src
                        /retek/sample_profiles
```

Note: We highly recommend that clients have a separate server that is for production only. When the time comes for a production environment, create a prd directory, then copy the dev directory structure to your production server and use the <base_directory>/rdw10.0/prd as your 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>/rdw10.0 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	retek	dba	775
dev	rdw10dev	dev	775
dbasql	rdwdba -Db2, Oracle, Teradata	dba	774

For example: `cd <base_directory>/rdw10.0`
`chown -R rdw10dev dev`
`chgrp -R dev dev`
`chmod -R 775 dev, etc...`

Copy profiles for Retek users

For each of the Retek users, copy the profiles.

- 1 At the Unix prompt, change directories to
`<base_directory>/rdw10.0/retek/sample_profiles`.

In this directory, you can find a sample of the profile that needs to be set up for each Unix user account.
- 2 Copy the profile to the home directory of each user. For example,
`rdw_profile_td` should be copied to `.profile` in the home directory of the users that were created. Edit the `.profile` to correspond with your file system and Unix environment.
- 3 Do this for the `rdw10dev`, `rdw10md`, and `rdw10dm` users.
- 4 When the time comes to create a test and/or production schema, create `rdwtst`, `rdwprd` and other Unix users. A copy of the corresponding development user's profile is often a good place to start.

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.

Chapter 6 – RDW RDBMS installation and configuration on Oracle

General installation notes

This release of RDW incorporates an optional interface with Retek Merchandising System (RMS) 10.0. This interface has been named the Data Warehouse Interface (DWI) and its release number will coincide with the RDW 10.0 product. Chapter 2 and 3 provide instructions on setting up the DWI 10.0 environment.

Note: You should become familiar with the Retek Data Warehouse development environment before setting up a production system.

Oracle Concepts for the Retek Data Warehouse

Oracle supports table level partitions or range partitions. Partitioning is a strategy that you should consider for your compressed datamarts to enhance performance. To take full advantage of the Oracle partitioning feature, you should partition large historical tables. To determine an appropriate partitioning strategy, 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 in RDW should be referenced when determining a partitioning strategy. Large non-compressed fact tables can be partitioned for ease of rolling off history. Before creating the RDW datamart tables in a production environment, determine your partitioning strategy. See the RDW Operations Guide to fully understand the Compression and Partitioning strategy that can be used when implementing the data warehouse on an Oracle platform.

Installation instructions

Oracle database configuration

The following instructions are recommended for the development and test environments only. Refer to your RDW sizing information to determine tablespace, table, index, sizes as appropriate for planning the production environment. The installation scripts provided will need to be modified with your production environment for your tablespace requirements, table and index storage parameters, and so on.

Create database and objects (tablespaces, users, and so on)

The scripts to create the RDW database, users, and tablespaces are in `<database_directory>/admin/rdw10/create` directory which was created in Chapter 5 Unix environment setup.

- Create an Optimal Flexible Architecture (OFA) compliant directory structure to store all Oracle files before creating the RDW database. The files have already been moved to the Unix server and untarred in a previous step.

Set up Parameter file:

- 1 Before creating the rdw10 database, the parameter file must be set up. A sample parameter file is provided.

See the file `<database_directory>/admin/rdw10/create/initrdw10.ora` for guidance on setting up the Oracle database parameters. This file should serve as a guideline for setting up the rdw10 database. It contains unformatted notes; therefore, it will not be valid to start a database without modification first.

- 2 Once this file has been updated and placed in the proper directory, alter and run the actual database creation scripts.

Create database:

- 1 Log in to Unix as Oracle. Change directories to `<database_directory>/admin/rdw10/create.`
- 2 Modify the sample database creation scripts `<database_directory>/admin/rdw10/create/crdb_ora.1` and `<database_directory>/admin/rdw10/create/crdb_ora.2` to contain the appropriate Unix file system paths, database name, number and size of redo logs, undo tablespace name and size, temporary tablespace name and size, and Oracle user password.

Note: These scripts assume that the RDW database name is rdw10, so change the path names and datafile names to reflect the actual RDW database name if necessary.

- 3 The default sizing of the tablespaces in these create scripts assume they are being used to set up a development environment
- 4 Log into SQL*Plus as sysdba: `> sqlplus '/ as sysdba'`, execute the first create script, and check the log for errors:
`SQL> @crdb_ora.1`
- 5 If the script executed properly, execute the second script and check the log for errors:
`SQL> @crdb_ora.2`
- 6 Review the log files for errors, if the script executed properly, continue with the database object creation. If not, make the appropriate corrections and rerun the necessary portions.

Database connectivity

Verify that you have connectivity and the necessary Net Services components on both the server and client machines. Samples are located in the *Sample Oracle Net Services Files for the Server* section in Appendix A.

Create tablespaces:

The directory `<database_directory>/admin/rdw10/create` contains scripts for creating the tablespaces needed by the RDW application.

- 1 Modify the script, `crtbspc_ora.sql` to include the correct Unix file system paths for the datafiles, as well as proper sizing/storage parameters for the environment being set up. The default sizing of the tablespaces in this create script assumes they are being used to set up a development environment.
- 2 The tablespace script makes the same assumptions regarding the database name (rdw10), so again, change as appropriate.
- 3 Connect to SQL*Plus as `system` and execute the following script:

```
SQL> crtbsp_ora.sql
```

- 4 Review the log file, `crtbsp.log`, for errors. If any errors occurred make the appropriate corrections and rerun the necessary portions.

Create RDW users:

- 1 Modify the script `crusers_ora.sql` by changing the passwords.

This script can then be used to create the default RDW development Oracle users and grant the appropriate system privileges. Refer to the Development Database Schema Implementation section in Appendix C for descriptions of each user.

- 2 Once edited, log in to SQL*Plus as `system` and execute the following script:

```
SQL>@crusers_ora.sql
```

- 3 Review the `crusers_ora.lst` file to verify that the users were all created successfully.
- 4 To secure the passwords in this script, change the privileges on the script by entering:

```
>chmod 700 crusers_ora.sql
```

Create RDW Datamart-level tables

- 1 Change directories to `<database_directory>/admin/rdw10/ddl/dm`.
- 2 Log in to SQL*Plus as `rdw10dm` and execute the following script:

```
SQL>@rdw10dm_ora.sql
```

This script creates the RDW datamart-level tables and calls subsequent scripts in the same directory. 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.

- 3 Review the `rdw10dm_ora.lst` file to verify that the objects were all created successfully.

Create MicroStrategy Repository

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 the following export files.

- 1 Change directories to `<database_directory>/admin/rdw10/ddl/md/ora`.
- 2 The import command will import the metadata objects using the `rdw10md_ora.dmp` file.

Import command

```
imp system/password parfile=rdw10md_ora.par
```

- 3 Review the `rdw10md_ora.log` file for errors and make appropriate corrections if needed and rerun.

To install the MicroStrategy metadata, complete the RDW Middle Tier installation documented in the RDW Middle Tier Installation Guide.

Set up RDW user permissions

- 1 Change directories on the Unix server to `<base_directory>/rdw10.0/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 what user should execute the script, when to run them and additional details about the scripts.
 - 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.
 - 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.
 - 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 Connect to SQL*Plus as `rdw10dm` and execute the following script to setup permissions to the datamart objects:

```
SQL>@gs_rdw10_dm_ora.sql
```

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

Note: The `invalids` script may need to be run several times to validate all of the interrelated objects.

- 3 Verify the scripts executed correctly by reviewing the following files:

```
g_rdw100_dm.lst, s_rdw100_dm.lst
```

- 4 Connect to SQL*Plus as `rdw10md` and execute the following script to setup permissions to the MicroStrategy database objects:

```
SQL>@gs_rdw10_md_ora.sql
```

- 5 Verify the scripts executed correctly by reviewing the following files:

```
g_rdw100_md.lst, s_rdw100_md.lst
```

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. You can do this by querying the `dba_objects` table looking for objects with a status of INVALID and recompiling the invalid objects. 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 RIB ETL

- 1 Log in to the Unix server with the `rdw10dev` account.
- 2 Change directories to `<base_directory>/rdw10.0/dev/rfx/etc`.
- 3 Modify the `config.env` script to match your environment:
 - a Change the `DBNAME` variable to the name of RDW database.
 - b Change the `DB_ENV` variable to `ORA`.
 - c Change the `DM_OWNER` variable to the username of the RDW Data Mart schema owner.
 - d Change the `BA_OWNER` variable to the username of the RDW batch user.
 - e Change the `LOAD_TYPE` to direct or conventional based on the requirements for SQL Loading. Refer to the Operations Guide Chapter Program Flow Diagrams for more details.
- 4 Continue on to Chapter 9, Default Data Population.

Chapter 7 –RDBMS installation and configuration on DB2

General installation notes

This release of RDW incorporates an optional interface with Retek Merchandising System (RMS) 10.0. This interface has been named the Data Warehouse Interface (DWI) and its release number will coincide with the RDW 10.0 product. Chapters 2 and 3 provide instructions on setting up the DWI 10.0 environment.

Note: You should become familiar with the Retek Data Warehouse development environment before setting up a production system.

Minimum requirements for DB2 database partitions and other configuration options, depending on transaction throughput desired:

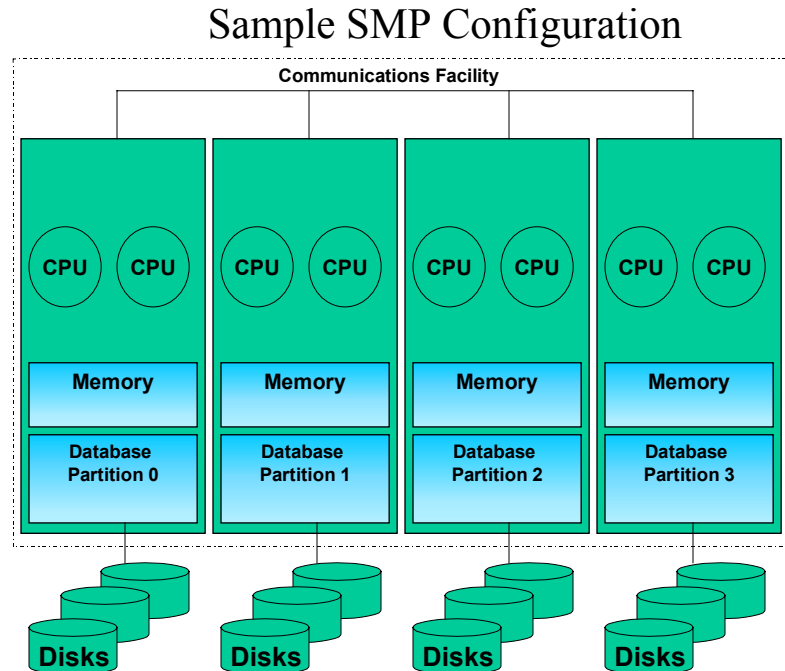
Hardware	Minimum	Recommendations are site specific
Multi-processors	8	12+
DB2 Database Partitions	4 nodes	6+
Memory	8 GB	12GB+

Note: The minimum recommended memory is 1 GB of memory per CPU. You most likely will see significant performance benefits with more than 1 GB of memory per CPU. All production sites are unique.

DB2 Concepts for the Retek Data Warehouse

The default DB2 installation will assume the minimum configuration of eight CPUs, four database partitions, and eight GB of memory running on a symmetric multi-processor (SMP) server. All installation scripts reflect this configuration. If you have a configuration other than the default one, modify the scripts to suit your environment.

Sample configuration on SMP:

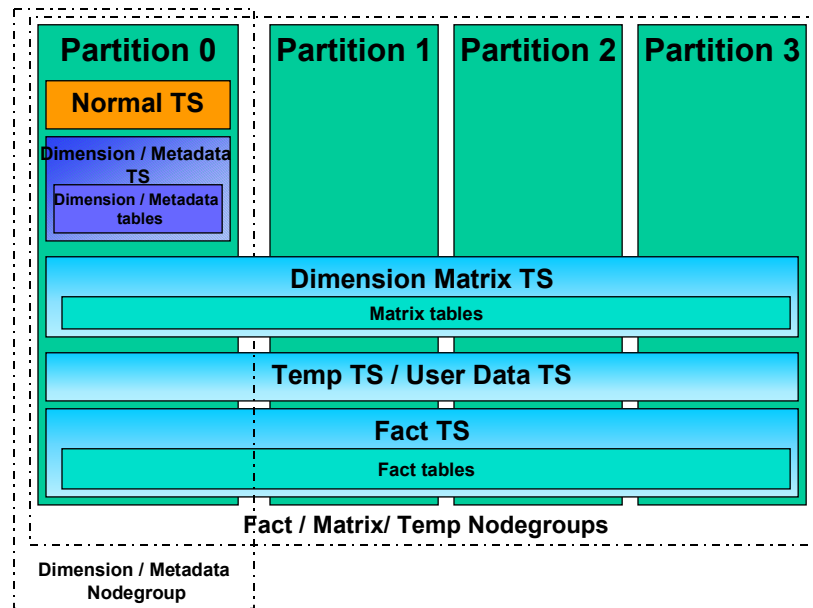


As an alternative, additional CPUs, partitions, memory and/or multiple symmetric multi-processor -SMP servers clustered or massively parallel processing – MPP server should be considered and carefully planned for your data warehouse environment. At this time an MPP configuration is not supported. If you are considering an MPP configuration, contact Retek for an estimated timeline for future support of an MPP architecture.

Tablespace placement should also be considered before creating your production system. This is a typical tablespace layout on four partitions with two node groups. By adding partitions, you can gain parallel processing power and scalability. The dimension and metadata tablespaces and tables are located in the first partition. The fact, temp, and user tablespaces and tables are spread across all the nodes. By spreading your large fact tables across all your partitions, you can perform your queries in parallel across all the partitions defined in that group. All installation scripts are based on this default configuration. If you have a different configuration modify the installation scripts to suit your environment.

Default tablespace configuration:

Default tablespace layout



Installation instructions

DB2 database configuration

We recommend the following instructions for development and test environments only. Refer to your RDW sizing information to determine tablespace, table, index, sizes as appropriate for planning the production environment. The installation scripts provided will need to be modified with your production environment for your tablespace requirements, table and index storage parameters, and so on.

Create database and objects (tablespaces/containers, users, and so on)

The scripts to create the RDW database, users, and tablespaces/containers are in Unix subdirectories of the `<database_directory>/admin/rdw10/create` directory. The files to create your database base directory structure have already been moved to the Unix server and untarred in chapter 5.

Create database:

- 1 Modify the sample database creation scripts. The following four scripts will need to be modified to contain the appropriate Unix file system paths, database name, number and size of logs, and tablespace size.

These scripts create a four-node database with the default tablespaces and buffer pools. The default sizing of the tablespaces in these create scripts assume they are being used to set up a development environment.

Several database configuration parameters will be modified in this script. Use these parameter settings as a baseline and modify as needed for your specific environment. The parameter settings are also listed as a guideline in the Appendix B section. The scripts are located in <database_directory>/admin/rdw10/create.

- crdb_db2.sql
- crtbspd_db2.sql
- crtbspi_db2.sql
- upd_cfg_db2.sh

Note: These scripts assume that the RDW database name is rdw10, so change the path names to reflect the actual database name if desired.

- 2 Log in as the DB2 instance owner. At the Unix prompt, change directories to <database_directory>/admin/rdw10/create.

```
>./run_crdb_db2.sh
```

A log file will be produced for each script. The log naming convention is <script_name>.log. Check the logs and verify there where no errors generated. If the script executed properly, continue with the database object creation. If not, make the appropriate corrections and rerun the necessary portions.

- 3 Verify the database configuration parameters are set correctly using the following command for each node.

```
>db2 get db cfg for rdw10 | more
```

Create RDW users:

- 1 Change directories to <database_directory>/admin/rdw10/create.
 The crusers_db2.sql script can then be used to create the default RDW development users and grant the appropriate system privileges. Refer to the Development Database Schema Implementation section for descriptions of each user.
- 2 Connect to rdw10 and execute the following script as the db2 instance owner:


```
>db2 connect to rdw10 user <db2 instance owner>
>db2 -tf crusers_db2.sql > crusers_db2.log
```

 Review the log file to verify that the users were all created successfully.
- 3 The crtbsp_privs_db2.sql script grants the datamart user the use of the fact data and index tablespaces. Execute the following script as the db2 instance owner.


```
>db2 connect to rdw10 user <db2 instance owner>
>db2 +p -tf crtbsp_privs_db2.sql > crtbsp_privs_db2.txt
```
- 4 A log file and txt file will be produced. Check the logs and verify there were no errors generated. If the script executed properly, continue with the database object creation. If not, make the appropriate corrections and rerun the necessary portions.

Create RDW Datamart-level tables

- 1 Change directories to <database_directory>/admin/rdw10/ddl/dm.

```
>db2 connect to rdw10 user rdw10dm
```

```
>./run_dm_db2.sh
```

This script creates the RDW datamart-level tables, views, primary keys, constraints and indexes. Review the logs for errors. Four logs will be generated in the same directory.

Create MicroStrategy Repository

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 the following export files.

- 1 Change directories to <database_directory>/admin/rdw10/ddl/md/db2.
- 2 The import script will call subsequent files in the same directory and create the metadata objects.

File to import

```
><database_directory>/admin/rdw10/ddl/md/db2/rdw10md_db2.sql
```

```
><database_directory>/admin/rdw10/ddl/md/db2/mstr7_db2.trg
```

File to start import

```
>db2 connect to rdw10 user rdw10md
```

```
>db2 -tf rdw10md_db2.sql > rdw10md_db2.log
```

```
>db2 -tvf mstr7_db2.trg > mstr7_db2.log
```

- 3 Review the log files for errors and make appropriate corrections if needed and rerun the necessary portions.

To install the MicroStrategy metadata, complete the RDW Middle Tier installation documented in the RDW Middle Tier Installation Guide.

Set up RDW user permissions

- 1 Change directories on the Unix server to <base_directory>/rdw10.0/dbasql.

This directory contains scripts that set up all necessary synonyms/aliases and grants for the RDW environment.

- The header of each file contains information regarding what user should execute the script, when to run them, and additional details about the scripts.
- All scripts have been set up to reference the default RDW usernames, so verify that they are correct and change to suit your environment if necessary.
- All scripts will create a SQL file with the necessary SQL statements, then the SQL file will be executed and output to a log file. The command syntax must be exactly as shown below.

- 2 Log in to the Unix server with the rdw10dm account.

- 3 Connect to rdw10 as rdw10dm and execute the following script to grant permissions to the datamart objects:

```
>db2 connect to rdw10 user rdw10dm
>db2 +p -tf gs_rdw10_dm_db2.dyn > gs_rdw10_dm_db2.sql
```

- 4 Review the log file, `gs_rdw10_dm_db2.log`, to verify the script executed successfully.
- 5 Log in to the Unix server with the `rdw10md` account.
- 6 Connect to rdw10 as `rdw10md` and execute the following script to grant permissions to the MicroStrategy metadata objects:

```
>db2 connect to rdw10 user rdw10md
>db2 +p -tf gs_rdw10_md_db2.dyn > gs_rdw10_md_db2.sql
```

- 7 Review the log file, `gs_rdw10_md_db2.log`, to verify the script executed successfully.

Note: These scripts are also used for maintenance. If you need to drop or add a table in the database, you need to 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 these scripts to alter the usernames for the environment you are creating.

Configure RIB ETL

- 1 Log in to the Unix server with the `rdw10dev` account.
- 2 Change directories to `<base_directory>/rdw10.0/dev/rfx/etc`.
- 3 Modify the `config.env` script to match your environment:
 - a Change the `DBNAME` variable to the name of RDW database.
 - b Change the `DB_ENV` to `DB2`.
 - c Change the `DM_OWNER` variable to the username of the RDW Data Mart schema owner, `RDW10DM`.
 - d Change the `BA_OWNER` variable to the username of the RDW batch user, `RDW10DEV`.
 - e Change the `LOAD_TYPE` to direct or conventional based on the requirements for data Loading. Refer to the Operations Guide Chapter Program Flow Diagrams for more details.
 - f Change the `SCHEDULE_TYPE` to sequential or parallel based on your batch processing schedule. Refer to the Operations Guide Chapter Program Flow Diagrams for more details.
- 4 Continue on to Chapter 9, Default Data Population.

Chapter 8 – RDBMS installation and configuration on Teradata

General installation notes

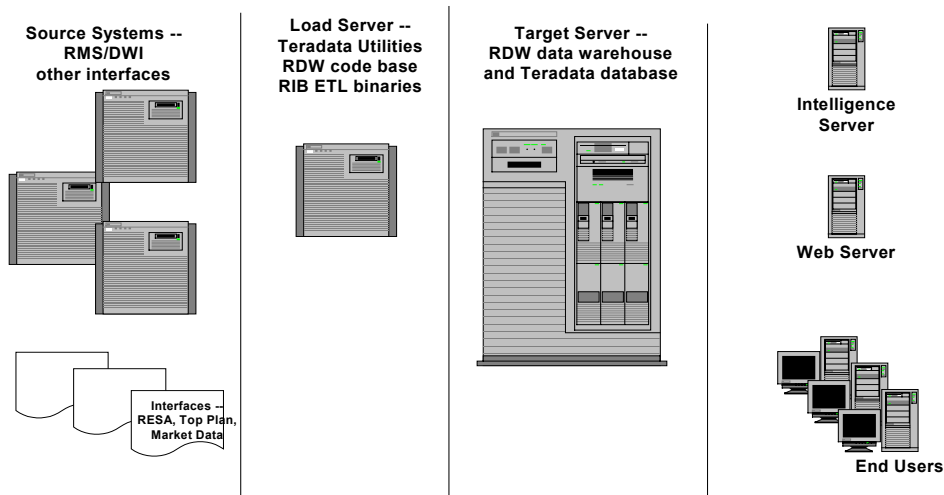
This release of RDW incorporates an optional interface with Retek Merchandising System (RMS) 10.0. This interface has been named the Data Warehouse Interface (DWI) and its release number will coincide with the RDW 10.0 product. Chapter 2 and 3 provides instructions on setting up the DWI 10.0 environment.

Note: You should become familiar with the Retek Data Warehouse development environment before setting up a production system.

Teradata Concepts for the Retek Data Warehouse

The Teradata environment has another component in addition to the source and target servers. A load server will be part of the environment, which will hold the Teradata Utilities, RDW code base, and the RIB ETL; and will connect to the source and target server via ODBC connections. The Teradata server is the target server and is where the RDW data warehouse is located.

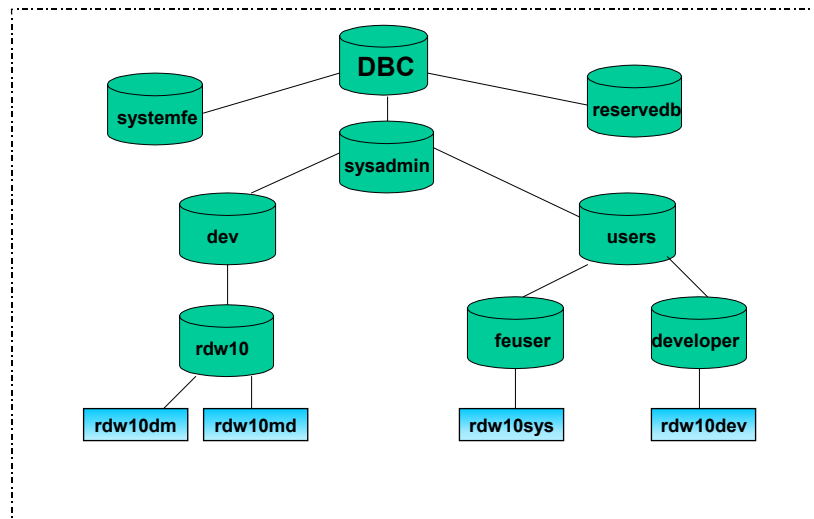
RDW Architecture on Teradata



This installation assumes that the sysadmin and systemfe have already been created. Teradata also recommends creating a reservedb, which contains approximately 15 to 20% of the database space for backup spool space.

Default RDW configuration:

Sample RDW Teradata Database Configuration



Installation instructions

Teradata database configuration

We recommend the following instructions for development and test environments only. Refer to your RDW sizing information to determine table, index, sizes as appropriate for planning the production environment.

Create database and objects

The scripts to create the RDW database and users are in the `<database_directory>/admin/rdw10/create` directory which was created in an earlier step.

The files to create your database base directory structure have already been moved to the Unix server and untarred in chapter 5.

Create database and users:

- 1 Log in to Unix as the Teradata user specified in Chapter 5. Change directories to `<database_directory>/admin/rdw10/create`.
- 2 Modify the sample database creation scripts `<database_directory>/admin/rdw10/create/crdb_td.sql` to contain the appropriate database name, perm and spool size. The default sizing of the databases in these create scripts assume they are being used to set up a development environment. These scripts can then be used to create the default RDW development users and grant the appropriate system privileges. Refer to the Development Database Schema Implementation section for descriptions of each user. These scripts assume that the RDW database name is rdw10, modify if necessary.

- 3 Modify the logon in the script, `run_crdb_td.sh`, to include the correct password for sysadmin.

```
>bteq <run_crdb_td.sh> run_crdb_td.log
```

A log file will be produced – `run_crdb_td.log`. Check the log and verify there where no errors generated. If the script executed properly, continue with the database object creation. If not, make the appropriate corrections and rerun only the necessary portions.

Create RDW Datamart-level tables

- 1 Change directories to `<database_directory>/admin/rdw10/ddl/dm`.
- 2 Modify the logon and password in the script `run_dm_td.sh`. Execute the script as the `rdw10dm` database user.

```
>bteq <run_dm_td.sh> run_dm_td.log
```

This script creates the RDW datamart-level tables, views, primary keys, constraints and indexes. Review the log file and verify there where no errors generated.

Create MicroStrategy Repository

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 the following export files.

- 1 Change directories to `<database_directory>/admin/rdw10/ddl/md/ter`.
- 2 Modify the script, `rdw10md_td.arc`, to include the Teradata server name and password. The script has been set up to reference the default RDW usernames, change to suit your environment if necessary.

Files to import

```
<database_directory>/admin/rdw10/ddl/md/ter/rdw10md_td.arc
```

```
<database_directory>/admin/rdw10/ddl/md/ter/mstr7_td.trg
```

- 3 Execute the following commands to start the import.


```
>arcmain < rdw10md_td.arc > rdw10md_td.log
```

```
>bteq .logon rdw10md,password < mstr7_td.trg > mstr7_td.log
```
- 4 Review the log files for errors and make appropriate corrections if needed and rerun.

To install the MicroStrategy metadata, complete the RDW Middle Tier installation documented in RDW Middle Tier Installation Guide.

Set up RDW user permissions

- 1 Change directories on the Unix server to <base_directory>/rdw10.0/dbasql.
This directory contains scripts that set up all necessary grants for the RDW environment.
 - The header of each file contains information regarding what user should execute the script, when to run them, and additional details about the scripts.
 - All scripts have been set up to reference the default RDW usernames, so verify that they are correct and change to suit your environment if necessary.
- 2 At a Unix Prompt enter:


```
>bteq .logon rdw10dm,password < \
      gs_rdw10_dm_td.dyn> gs_rdw10_dm_td.log
      >bteq .logon rdw10md,password < \
      gs_rdw10_md_td.dyn > gs_rdw10_md_td.log
```
- 3 Review the log files for errors and make appropriate corrections if needed and rerun the necessary portions.

Note: These scripts are also used for maintenance. If you need to drop or add a table in the database, you need to 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 these scripts to alter the usernames for the environment you are creating.

Configure RIB ETL

- 1 Log in to the Unix server with the rdw10dev account.
- 2 Change directories to <base_directory>/rdw10.0/dev/rfx/etc.
- 3 Modify the config.env script to match your environment:
 - a Change the DBNAME variable to the name of Teradata server.
 - b Change the DB_ENV to TER.
 - c Change the DM_OWNER variable to the username of the RDW Data Mart schema owner, RDW10DM.
 - d Change the BA_OWNER variable to 'DEVELOPER', the default database for the RDW batch user.
- 4 Continue on to Chapter 9, Default Data Population.

Chapter 9 – Default data population

ETL installation and configuration

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

Data population

Some of the RDW tables require one or more default records before setting up the batch cycle. Log in to a Unix session as rdw10dev. If you are setting up a test or production environment, log in with the appropriate Unix account – rdw10tst for the test system, rdw10prd for the production system.

If the account is set up with the proper profile information in the .profile, the environment variables will be set up by logging into the appropriate Unix user. The key environment variables for running RIB-ETL programs are: \$MMUSER, \$PASSWORD, and \$MMHOME.

Note: All SQL scripts, error and log files are encoded UTF-8. Any changes or modifications to these SQL scripts should be made with your UNIX session set to UTF8. Refer to your operating system documentation for more details on your session settings.

Populate datamart default data

- 1 Log in to a Unix session as rdw10dev. Change the MMUSER and PASSWORD variables to the data mart owner, rdw10dm and the appropriate password. Verify the RIB ETL executable is in the path of your Unix session.
- 2 Modify the config.env file in the
<base_directory>/rdw10.0/dev/rfx/etc directory. Change the BA_OWNER to rdw10dm, the data mart owner.
- 3 Change directories on the Unix server to
<base_directory>/rdw10.0/dev/install. This directory contains the code for all default data installation scripts.
- 4 At the Unix prompt, enter:


```
>default_load.ksh
```

This script creates the date_key.txt file in
<base_directory>/rdw10.0/dev/rfx/etc and executes the following scripts:

 - alter_dt_fmt.sql - alters the current session's date format (Oracle only)
 - 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.
 - `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`
- 5 Change directories on the Unix server to `<base_directory>/rdw10.0/dev/log`. Review the log file that was created.
 - 6 Change directories on the Unix server to `<base_directory>/rdw10.0/dev/error`. Review the error files that were created for `default_load` and `static_load`.
 - 7 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 or a 13 period time calendar. The client must populate these tables according to the client's specifications and business requirements. To populate the time dimension with 454 calendar time from RMS proceed to section A 'Extract 454 Time from RMS'. To populate time for 454 or 13 period standalone proceed to section B 'Create Standalone Time'. For information on the tables loaded for the Time dimension refer to the RDW Data Model documentation.

A. Extract 454 Time from RMS

- 1 Change directories to `<base_directory>/dwi10.0/unix/dev/install` (on RMS server).
- 2 Modify the variable `l_path` in the `extract_time.sql` script to reference the `UTL_FILE` directory specified in the RMS database parameter file.
- 3 Login in to SQL*Plus as RMS schema owner (`rmsdev100` or the equivalent) to extract the time data out of the RMS database.

```
SQL>@extract_time.sql
```

This script generates another file called time_454.txt and wkday.txt located in the utl_file_dir directory specified in your RMS database parameter file.

- 4 Move the text files to your base_directory>/rdw10.0/dev/install directory.

You are limited to the time originally extracted from RMS. If time loaded in RMS has changed, then extract the additional time from RMS again and repeat the steps above.

- 5 Proceed to section C ‘Create Time in RDW’ to load the RMS time into the Data Warehouse.

B. Create Standalone time

Modify the sample text file that will serve as the data to populate the time calendar dimension with appropriate time for your environment. If you are implementing a 454 time calendar, modify the time_454.txt. If you are implementing a 13 period time modify the time_13.txt file with your time calendar data. A sample of these text files has been provided in <base_directory>/rdw10.0/dev/install. See the RDW Operations Guide Chapter 9 ‘API Flat File Specifications’ for instructions on how to create the text file.

C. Create Time in RDW

If you need to add more time data, modify the parameters below for the new time period and run the module again. Make sure that the new time being created does not overlap existing time.

- 1 Change directories to <base_directory>/rdw10.0/dev/install.
- 2 Login to a Unix session as rdw10dev. Change the MMUSER and PASSWORD variable to the batch user, rdw10dev and the appropriate password. Verify the RIB ETL executable is in the path of your Unix session.
- 3 Modify the config.env file in the <base_directory>/rdw10.0/dev/rfx/etc directory. Verify the LOAD_TYPE is set to conventional. Change the BA_OWNER to the batch user, rdw10dev.
- 4 Compile the module crtmdm.c with a standard ANSI C compiler. 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:


```
>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
```

- 5 Execute the `cr_time.ksh` module. At the Unix prompt enter:

```
>cr_time.ksh
```

This script will prompt for your calendar. Choose 1 for 454 time and 2 for 13 period time.

At the prompt enter the 4 digit year for the beginning and ending of your time calendar:

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

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

One text file will be generated in the install directory for each dimension table.
- 6 At the Unix prompt, enter:

```
>time_load.ksh
```

```
>time_trnsfrm_load.ksh
```
- 7 Change directories on the Unix server to

```
<base_directory>/rdw10.0/dev/log
```

. Review the logs file that was created.
- 8 Change directories on the Unix server to

```
<base_directory>/rdw10.0/dev/error
```

. Review the error files that were created for `time_load` and `time_trnsfrm_load`.
- 9 Refer to the RDW Operations Guide for more information on the log files and error files.
- 10 Modify the `config.env` file in the

```
<base_directory>/rdw10.0/dev/rfx/etc
```

 directory. Change the `LOAD_TYPE` to what has been determined for your environment. Refer to the RIB ETL Programmers Guide for more details.

Populate static dimension data

There are several dimension tables in RDW that must be loaded once 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>/rdw10.0/dev/install` directory, but the client must populate these tables according to individual specifications and business requirements.

The tables to be populated and loaded 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 Change directories on the Unix server to
`<base_directory>/rdw10.0/dev/install`. This directory contains the code for all default data install scripts.
- 2 Login to a Unix session as `rdw10dev`. Change the `MMUSER` and `PASSWORD` variable to the batch user, `rdw10dev` and the appropriate password. Verify the RIB ETL executable is in the path of your Unix session.
- 3 The script `timelfldm.ksh` loads the following Time ‘Like for Like’ tables that are used exclusively by the MicroStrategy reports:
 - a `time_last_yr_by_day_lfl_dm`
 - b `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>/rdw10.0/dev/log`. Review the log file that was created.
- 6 Change directories on the Unix server to
`<base_directory>/rdw10.0/dev/error`. Review the error files that were created.
- 7 This script 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>/rdw10.0/dev/log`. Review the log file that was created.
- 9 Change directories on the Unix server to
`<base_directory>/rdw10.0/dev/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.

Appendix A – Oracle information

Compile scripts for DWI Pro*C modules

The DWI compile scripts are located in the
`<base_directory>/dwi10.0/retek/bin.`

The only script that may require manual modification is `setplatflags.ksh`, which contains platform-specific compile flags.

Rdwhcomp

This script is used to compile code for development or testing purposes. It compiles a single batch program in the current working directory. It calls `setplatflags.ksh` to set platform-specific flags, then calls `makefile batch.mk`.

Rdwfcomp

This script is used to compile and migrate production code. It compiles a single batch program in the current working directory, clears out the `.c` and `.o` files, and copies the executable into the product binary directory `$MMHOME/oracle/proc/bin`. It calls `setplatflags.ksh` to set platform-specific flags, then calls `make batch.mk`, unsetting the debug flags as it does so.

Rdwfcall

This script simply calls `rdwfcomp` for all batch modules in the current working directory. It is most useful for migrating changes in the main source directory `$MMHOME/oracle/proc/src`.

Rdwlcomp

This script is used to compile the batch libraries. It accepts one argument: `dwi10`, depending on which system is being built. This should ONLY be invoked when the current working directory is the library source directory – `$MMHOME/oracle/lib/src`. It calls `setplatflags.ksh` to set platform-specific flags, then calls `make library.mk`, passing in the argument to the script as a target. By default, it also sets debug flags so that libraries can be debugged along with batch program code. This should be unset when compiling production code.

setplatflags.ksh

This script sets platform-specific compile flags to be used in `batch.mk` and `library.mk`. Depending on your platform, more flags may need to be added to this script. Two variables are set here. `PROC_PLATFORM` contains flags to be sent to `proc` when precompiling the Pro*C code. `CC_PLATFORM` contains flags to be sent to `cc` when compiling the C code. Refer to Oracle's sample Pro*C makefiles for suggestions on compile flags for your platform.

Makefiles

The makefiles should not require any modifications.

batch.mk

This makefile contains a suffix rule to compile an executable from a .pc source file. It imports a number of variables from Oracle Pro*C's default environment makefile (\$ORACLE_HOME/precomp/lib/env_precomp.mk), including the name of the client-shared library (libclntsh.so), which contains the code that allows a batch program to communicate with the database. It then sets variables defining compile options and paths.

The actual compilation of a batch program consists of three steps:

- 1 Precompile the .pc file, generating a .c file.
- 2 Compile the .c file, producing a .o file.
- 3 Link the .o file with Oracle's library and Retek's library to produce the executable.

library.mk

This makefile contains rules to compile standard library modules for DWI. It imports a number of variables from Oracle Pro*C's default environment makefile (\$ORACLE_HOME/precomp/lib/env_precomp.mk), including the name of the client-shared library (libclntsh.so), which allows a program to communicate with the database. It then sets variables defining compile options and paths.

The actual compilation of a library module consists of two steps:

- 1 Precompile and compile each .pc module, producing a .o file for each (if they haven't already been compiled).
- 2 Link the .o files into the library archive module (\$MMHOME/oracle/lib/bin/retex.a).

Sample Oracle Net Services Files for the Server

Net Services Setup

- 1 Install the necessary Net Services components on both the server and client machines. If Net Services already exists and is operational on the server, manually alter the files \$ORACLE_HOME/network/admin/listener.ora and \$ORACLE_HOME/network/admin/tnsnames.ora according to the Oracle 9i documentation.
- 2 Check to see whether the Net Services listener is running by logging into Unix as the user oracle and then issuing the following command:
lsnrctl stat
- 3 If the listener is running, stop it by issuing the following command:
lsnrctl stop
- 4 Start the listener by issuing the following command:
lsnrctl start

Modify the client-side tnsnames.ora file

- 1 Edit the \$ORACLE_HOME\network\admin\tnsnames.ora file on the client to match the settings on the Unix server.
- 2 After you have altered this file on one PC and have tested the connectivity, copy this file to other PCs, if necessary.

The tnsnames.ora file for client PCs may be on a LAN. In this case, it is not necessary to copy the file to each PC.

Samples of a typical listener.ora and the tnsnames.ora files are below.

listener.ora

```
listener =
  (ADDRESS_LIST=
    (ADDRESS= (PROTOCOL=IPC) (KEY=rettek.world))
    (ADDRESS= (PROTOCOL=IPC) (KEY=rettek))
    (ADDRESS= (PROTOCOL=tcp) (HOST=<IP
address>) (PORT=1521))
  )
  SID_LIST_LISTENER =
    (SID_LIST=
      (SID_DESC=
        (GLOBAL_DBNAME= <IP address.network domain>)
        (SID_NAME=rettek)
        (ORACLE_HOME=<oracle home directory>)
      )
    )

  (SID_DESC =
    (SID_NAME - extproc)
    (ORACLE_HOME=<oracle home directory>)
    (PROGRAM=extproc)
  )
)

STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER=ADMIN
TRACE_DIRECTORY=$ORACLE_HOME/network/trace
```

tnsnames.ora

```
retек.world =  
  (DESCRIPTION=  
    (ADDRESS_LIST =  
      (ADDRESS =  
        (PROTOCOL=tcp)  
        (host=<IP address>)  
        (Port=1521)  
      )  
    )  
    (CONNECT_DATA=  
      (SID = retек) (srvr=dedicated)  
      (GLOBAL_NAME = retек.world)  
    )  
  )
```

Appendix B – DB2 information

Sample database configuration parameters on DB2

Use the following parameters as a starting point for DB2 databases running RDW. The parameters can be modified for your data warehouse and your environment.

Note: There may be differences per specific environments, as these are the parameters specifically used with DB2 version 7.2.

Parameter	Value
BUFFPAGE	60000 (general guideline: 20000 * # of nodes)
CATALOGCACHE_SZ	512
CHNGPGS_THRESH	80
DBHEAP	8000
LOGBUFSZ	512
LOGFILSIZ	10000
LOGPRIMARY	10
LOGSECOND	50
LOCKLIST	1024
MAXAPPLS	100
MAXFILOP	128
NEWLOGPATH	Designate a separate log path for each node.
NUM_IOCLEANERS	12 (general guideline: # of CPUs)
NUM_IOSERVERS	12
SOFTMAX	700
SORTHEAP	1024
APPLHEAPSZ	4096

Appendix C – Additional information

Development Database Schema implementation

The following figure shows the DWI/RMS and RDW development schemas that are referred to in this installation guide.

