Oracle® Rdb for OpenVMS

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Oracle® Rdb for OpenVMS

Installation Guide

Release 7.2.5.3

July 2013

Oracle Rdb Installation Guide, Release 7.2.5.3 for OpenVMS I64 and OpenVMS Alpha

Copyright © 1984, 2013 Oracle Corporation. ***All rights reserved.***

Primary Author: Rdb Engineering and Documentation group

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error−free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency−specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227−19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail−safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, Oracle Rdb, Hot Standby, LogMiner for Rdb, Oracle SQL/Services, Oracle CODASYL DBMS, Oracle RMU, Oracle CDD/Repository, Oracle Trace, and Rdb7 are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third−party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third−party content, products, or services.

Contents

Preface

The Oracle Rdb software is a general−purpose database management system based on the relational data model. This manual uses the name Oracle Rdb to refer to current and previous versions of the software.

Purpose of This Manual

This manual describes how to install Oracle Rdb Release 7.2 on the HP OpenVMS for Alpha and HP OpenVMS Industry Standard 64 for Integrity Servers operating systems. You do not have to install a previous version of Oracle Rdb before installing Oracle Rdb Release 7.2.

Intended Audience

Read this manual if you are responsible for:

- Planning the installation of Oracle Rdb and preparing your system (see Chapter 1)
- Installing Oracle Rdb (see Chapter 2)
- Changing your system by adjusting parameters, startup and shutdown files, and privileges required for running Oracle Rdb (see Chapter 3)
- Configuring your Oracle Rdb system to allow remote database access (see Chapter 4)

To install the software, you must:

- Be familiar with VMSINSTAL, the command procedure used to install software products in the OpenVMS environment. For details on VMSINSTAL, see the OpenVMS system management documentation.
- Have access to the SYSTEM account on your machine or to an account with the user privilege SETPRV.

Document Structure

This manual consists of the following chapters and appendixes:

Chapter 1	Explains how to plan the installation and prepare your system.
Chapter 2	Explains how to install the Oracle Rdb software and run the Installation Verification Procedure (IVP).
Chapter 3	Explains procedures to follow after the installation of Oracle Rdb completes successfully.
Chapter 4	Explains how to configure your Oracle Rdb system to allow remote database access.
Appendix A	Shows a sample installation of Oracle Rdb.
Appendix B	Discusses the correlation between OpenVMS and Oracle Rdb security.

Related Manuals

The OpenVMS documentation set contains detailed information and guidelines for installing software on your OpenVMS system and for learning about related system management tasks.

The Oracle Rdb Release Notes might contain information needed to install Oracle Rdb. Read this document before starting the Oracle Rdb installation.

The Oracle SQL/Services Release 7.3.1.1 Installation Guide describes how to install the SQL/Services component of Oracle Rdb.

References to Products

The Oracle Rdb documentation set to which this manual belongs often refers to the following products by their abbreviated names:

- OpenVMS I64 refers to the HP OpenVMS Industry Standard 64 for Integrity Servers.
- OpenVMS refers to the OpenVMS Alpha and OpenVMS I64 operating systems.
- Oracle Rdb refers to Oracle Rdb for OpenVMS Alpha and OpenVMS I64 software. Release 7.2 of Oracle Rdb software is often referred to as Release 7.2 or V7.2.
- The SQL interface to Oracle Rdb is referred to as SQL. This interface is the Oracle Rdb implementation of the SQL standard adopted in 1999. This standard is referred to as the ANSI/ISO SQL standard or SQL:1999. SQL:1999 supersedes the SQL92 standard.
- In Oracle Rdb documentation, the terms release and version (and their abbreviations) are sometimes used interchangeably. You may see, for example, references to version 7.2, V7.2, and release 7.2.
- Oracle CDD/Repository software is referred to as the dictionary, the data dictionary, or the repository.
- Oracle ODBC Driver for Rdb software is referred to as the ODBC driver.
- Oracle Trace for OpenVMS software is referred to as Oracle Trace.
- Hewlett−Packard Company is referred to as HP.
- DECnet and DECnet−Plus refer respectively to HP DECnet for OpenVMS and HP DECnet−Plus for OpenVMS. DECnet Phase IV is used interchangeably with DECnet in this document.
- UCX refers to HP TCP/IP Services for OpenVMS.

Chapter 1
Preparing to Install Oracle Rdb

This chapter discusses the preparations and requirements necessary for installing Oracle Rdb.

1.1 Oracle Rdb License Options and Packaging

The Oracle Rdb installation kit includes the following products:

- Oracle Rdb provides the following:
    - Interactive SQL utility, including data definition as well as data manipulation
    - Support for dynamic SQL
    - Oracle RMU, the Oracle Rdb management utility
    - Support for the execution of previously developed application
- Programmer for Rdb (Rdb Compilers) includes all Oracle Rdb compilers, including, for example:
    - SQL precompiler
    - SQL Module Language processor
    - RDBPRE precompilers
    - RDML precompiler
- Hot Standby installs the files and images necessary to use the Hot Standby capability, which enables you to replicate an Oracle Rdb database at a remote standby site. If a node, cluster, or master database fails, the standby database can take over application processing. For more information on Hot Standby, refer to Section 3.8.4.
- Power Utilities installs files and images necessary for using Oracle RMU's parallel processing capabilities for the RMU Backup and RMU Load commands. To use the parallel RMU Backup functionality, Oracle SQL/Services Release 7.2 or later must be installed on your system.

The following are also available with the Oracle Rdb installation kit. However, they are installed separately:

- Oracle SQL/Services, which can be purchased with any of the following network protocols:
    - DECnet
    - TCP/IP
    - Novel NetWare
    - OCI Services for Oracle Rdb
- Oracle ODBC Driver for Rdb

1.2 Prerequisite Software

This section discusses the software you must have installed on your system before installing Oracle Rdb. This section also includes information about software that you can use with Oracle Rdb. Information about compatible products and their required version numbers is available at the following URL:

```
http://www.oracle.com/technetwork/products/rdb/index.html
```

1.2.1 Operating System Requirements

Oracle Rdb Release 7.2 requires one of the following OpenVMS environments:

- OpenVMS Alpha version 8.2 or later
- OpenVMS I64 version 8.2−1 or later

1.2.2 EPC$SHR.EXE Shared Image

Oracle Rdb requires that SYS$LIBRARY:EPC$SHR.EXE be installed as a shareable, protected image. This image is included with all OpenVMS installations, as well as with Oracle Trace, and should already be installed correctly. The Oracle Rdb installation procedure and startup procedure (RMONSTART72.COM) will verify that this image is installed correctly.

If SYS$LIBRARY:EPC$SHR.EXE is not found on your system, the installation or startup will fail.

To check that EPC$SHR.EXE is installed correctly, issue the following command:

```
$ INSTALL LIST SYS$LIBRARY:EPC$SHR.EXE
```

This should produce output similar to the following:

```
DISK:<SYSCOMMON.SYSLIB>.EXE EPC$SHR;3        Open Hdr Shar     Prot Lnkbl
```

1.3 Optional Software

Oracle Rdb Release 7.2 is compatible with many Oracle software products. These products include Oracle CDD/Repository, Oracle Trace, and Oracle Replication Option for Rdb.

Oracle Rdb Release 7.2 is also compatible with various standard programming languages that support the OpenVMS Calling Standard. Unless specifically mentioned, Oracle Rdb works with any supported version of these products. Take special note of the following points affecting optional software:

- Oracle CDD/Repository
  Oracle Rdb Release 7.2 for OpenVMS requires that Release 7.2 or later of Oracle CDD/Repository be installed. Use the Common Dictionary Operator (CDO) utility to see if the correct version of Oracle CDD/Repository is installed on your system.

  ```
  $ REPOSITORY OPERATOR
  Welcome to CDO V2.3
  The CDD/Repository V7.2 User Interface
  Type HELP for help
  CDO> EXIT
  ```

  See Section 1.7.3 for information on the order in which you install Oracle CDD/Repository and Oracle Rdb software.
- Replication Option for Rdb
  The Replication Option for Rdb is a separate installation from Oracle Rdb. See the Replication Option for Rdb Installation Guide for additional information.
- LSE
  If you want the Language−Sensitive Editor (LSE) template support for SQL statements, install LSE before installing Oracle Rdb. Oracle Rdb Release 7.2 is compatible with LSE version 4.7 or later.

  Note

  > ***The LSE templates provided with Oracle Rdb provide support only for SQL syntax through Release 4.7. The templates do not provide support for new and changed syntax after Release 4.7.***

Oracle Rdb Release 7.2 is compatible with many software products from HP, including COBOL, ACMS and HP DATATRIEVE.

1.4 Disk Space Requirements

The minimum storage requirement for installing Oracle Rdb for OpenVMS Alpha is 280,000 blocks; the minimum storage requirement for installing Oracle Rdb for OpenVMS I64 is 500,000 blocks.

To determine the number of available disk blocks on the current system disk, enter the following command at the DCL prompt:

```
$ SHOW DEVICE SYS$SYSDEVICE
```

The Oracle Rdb installation procedure provides files and images in specific directories on the system disk. These directories must exist for the installation to succeed. Logical names such as SYS$HELP and SYS$TEST are not translated by the installation procedure. If you have moved any SYS$COMMON directories to other devices to save space on your system disk, please be sure to re−create these directories on the system disk before installing.

1.5 Monitor Process Quota Requirements

When an Oracle Rdb monitor process (RDMMON) is started using the RMU Monitor Start command, the quota limits that the monitor process uses are determined as the largest of three factors:

- A hard−coded "minimum−necessary" value.
- The quota value from the user designated by the RDM$MON_USERNAME logical name (with a default value of "SYSTEM").
- The quota value from the process performing the startup.

The hard−coded minimum value for each monitor quota is shown in Table 1−1.

Table 1−1 Monitor Process Hard−Coded Minimum Quotas

Quota	Minimum Value
ASTLM	256
BIOLM	256
BYTLM	250000
DIOLM	256
ENQLM	1048575
FILLM	2048
PGFLQUOTA	250000
PRCLM	64
TQELM	256
WSEXTENT	512
WSQUOTA	512

These quota value minimums help prevent the monitor from being unable to open many large databases.

1.6 Database Server Process Quota Requirements

The various Oracle Rdb database server processes (ABS, ALS, LCS, LRS, RCS, and DBR) are started by the database monitor (RDMMON).

The database monitor process starts the server processes with quotas based on the quotas for the monitor. Each quota is determined as the larger of the monitor's quota and a hard−coded minimum value. If the monitor is started using a process or account (via the RDM$MON_USERNAME logical name) with quotas greater than the minimum, the monitor's quotas will be used. This provides the ability to increase quotas for the server processes beyond the minimum, if needed.

In general, the quota values should be adequate for all systems. In fact, some of the quota values have been chosen to be the maximum allowed OpenVMS value.

The hard−coded minimum value for each database server quota is shown in Table 1−2.

Table 1−2 Database Server Process Hard−Coded Minimum Quotas

Quota	Minimum Value
ASTLM	32767
BIOLM	32767
BYTLM	99999999
DIOLM	32767
ENQLM	1048575
FILLM	2048
PGFLQUOTA	99999999
PRCLM	100
TQELM	32767
WSEXTENT	32767
WSQUOTA	512

The database servers that are affected by the quota minimums are shown in Table 1−3.

Table 1−3 Database Server Processes

Name	Server
ABS	AIJ backup server
ALS	AIJ log server
DBR	Database recovery
LCS	AIJ log catchup server
LRS	AIJ log recovery server
RCS	Row cache server

1.7 Preparing Your System and the Installing Account

The following sections discuss the steps you must take and the requirements you must meet before installing Oracle Rdb Release 7.2.

1.7.1 Backup, Restore, and Recovery Operations with a New Version of Oracle Rdb

As a safety precaution, back up all Oracle Rdb databases, including Oracle Trace and CDD/Repository databases, with the RMU Backup command before installing Oracle Rdb Release 7.2.

Planning an appropriate conversion strategy and procedure for upgrading to a more recent or the most current release of Oracle Rdb depends on the version you are currently using and the version to which you want to upgrade.

Section 1.7.1.1 describes how to upgrade from Oracle Rdb Release 7.0, or 7.1 to Release 7.2 using the RMU Convert command.

You cannot convert databases earlier than Oracle Rdb Release 7.0 directly to Release 7.2.

If you have a database from release 6.0 or 6.1, you must first convert to release 7.0 or 7.1. See Section 1.7.1.1 for instructions to convert the intermediate database to Release 7.2.

If you have a database from release 3.0 through release 5.1, you must first convert it to release 6.0 or release 6.1, then convert that result to release 7.0 or 7.1. See Section 1.7.1.1 for instructions to convert the intermediate database to release 7.2.

1.7.1.1 Upgrading from an Oracle Rdb Prior Release to Release 7.2

If you are using a version of Oracle Rdb from release 7.0 or 7.1 and want to upgrade to Release 7.2, the general strategy is as follows:

1. Back up your databases.
   a. Use the RMU Close command to close the databases from user access.
   b. Use the SQL ALTER DATABASE statement to open the databases manually to limit user access and allow only operator access.
   c. Back up the databases using the RMU Backup command and perform a full backup of the databases.
   d. Disable the .AIJ file for each database, using the SQL ALTER DATABASE statement.
2. Install Oracle Rdb Release 7.2. After installing:
   a. Reset the DCL tables on each node of the cluster.
   b. Start the Oracle Rdb monitor process by executing RMONSTART72.COM on all nodes of your cluster. The installation automatically starts the monitor on the node from where you are installing.
3. Convert your databases using the RMU Convert command with the Commit qualifier.
   a. Use the SQL ALTER DATABASE statement to open the databases manually to limit user access and allow only operator access in combination with the RMU Open command with the

    Access=Restricted qualifier.
    b. Optionally, verify the integrity of the database or databases using the RMU Verify command (verify a database only if you suspect problems). If the RMU Verify command returns no error messages, the database integrity is sound.
    c. Use the SQL ALTER DATABASE statement to enable the .AIJ file for each database.

4. Use the RMU Backup command to back up the new databases.
    a. Optionally, use the RMU Dump command with the Backup_File qualifier to verify the integrity of the backup file for each database (only if you experience backup problems such as media errors).
    b. Use the RMU Close command to close the databases.
    c. Use the RMU Open command to open the databases for user access.

Always backup your databases before and after database conversions. Limit user access until all maintenance operations are complete and enable the .AIJ files before users access the databases.

1.7.2 Reverting to Release 7.0 or 7.1 from Release 7.2

If you have converted a prior version database to release 7.2 and have not committed the conversion by specifying the RMU Convert command with the Nocommit qualifier in the original database conversion, you can revert to the prior version by specifying the Rollback qualifier in a subsequent RMU Convert command. You can also commit the conversion permanently by specifying the Commit qualifier in a subsequent RMU Convert command.

Note

If you specified the Commit qualifier in the original database conversion operation or performed the RMU Convert command without specifying the Commit qualifier, the default conversion assumes that the Commit qualifier was specified and your database is permanently converted. You cannot roll back a conversion−committed database.

Because the .AIJ file format for a previous version is not compatible with a higher version, use the following procedure if you started using release 7.2 and enabled journaling and do not want to lose your updates committed under a previous version:

1. Run the RMU Convert command with the Rollback qualifier on your converted but not yet conversion−committed database. The RMU Convert command with the Rollback qualifier returns your database to its version before it was originally converted.
2. Return to Oracle Rdb Release 7.0 or 7.1 and install Release 7.0 or 7.1 again.
3. Perform a backup with an RMU Backup command on the reverted database. Backing up your database preserves the current contents of the database files, including all updates to the database while it was in its converted state.
4. Continue normal operations. Enable after−image journaling and start with a new, empty .AIJ file. Discard the .AIJ files created by release 7.2. These files are no longer useful after you have made a backup of the reverted database.

1.7.3 CDD/Repository Considerations

You must install Oracle Rdb before installing CDD/Repository Release 7.2. If you are also installing Oracle

CODASYL DBMS, the order of installation is Oracle Rdb first, then CDD/Repository, and finally Oracle CODASYL DBMS.

1.7.4 OpenVMS Privileges Required

VMSINSTAL is located in SYS$UPDATE, which is a restricted directory. To install Oracle Rdb, you must use an account that has the SETPRV privilege authorized. As one of its first actions, the VMSINSTAL command procedure grants all privileges except BYPASS to the process that invokes it. The VMSINSTAL command procedure succeeds only if the account has SETPRV privilege.

To check the default privileges of the installing account, log in and enter this command:

```
$ SHOW PROCESS/PRIVILEGES
```

If the installing account lacks the SETPRV privilege, you cannot install Oracle Rdb. You have two options:

- Ask your system manager to use the OpenVMS Authorize utility (AUTHORIZE) to modify the default privileges of the account to include the SETPRV privilege.
- Run AUTHORIZE and make the changes yourself, if the installing account has the SYSPRV privilege:

  ```
  $ SET DEFAULT SYS$SYSTEM
  $ RUN AUTHORIZE
  UAF> MODIFY <account−name>/PRIVILEGES=(SETPRV)
  UAF> EXIT
  ```

To activate the change in privileges, you must log out and log in again.

Note

When installing Oracle Rdb on systems with DECnet−Plus, the installation account must also have the NET$MANAGE identifier.

1.7.5 Process Account Password Must Not Be Locked

The installing account cannot have a locked password. If this is the initial installation of Oracle Rdb, the procedure creates an account called RDB$REMOTEnn (where nn is the version number). If the installing account has a locked password, the installation procedure is unable to automatically generate a password for this account, and aborts with the following message:

```
*******************************************************************

Error generating password for remote account.

*******************************************************************
```

To modify an account with a locked password, use the Authorize utility. You must have system privileges to use the Authorize utility.

```
$ RUN AUTHORIZE
UAF> MODIFY <account−name>/FLAGS=NOLOCKPWD
```

```
UAF> EXIT
```

1.7.6 Process Account Quotas Required

The account you use to install Oracle Rdb must have sufficient quotas to run the software. See Section 3.6 for minimum account quota values.

1.7.7 System Parameter Values Required

Installing Oracle Rdb requires minimum values for some system parameters. Depending on the kinds of programs and applications running at your site, you might need higher values for some settings. Table 1−4 lists the system parameter values required for installing Oracle Rdb.

Table 1−4 lists some parameters whose units are specified in pages. On OpenVMS systems, the size of a page can differ on different CPUs. With the exception of GBLPAGFIL, read the values in Table 1−4 as 512−byte pagelets, which are not CPU−specific. GBLPAGFIL values on OpenVMS systems are expressed in CPU−specific pages, typically 8192 bytes.

Table 1−4 Required Minimum System Parameter Values

System Parameter	Value
CHANNELCNT	A number larger than the largest FILLM used on the system
CLISYMTBL	512 pages (Necessary only during the installation procedure. If the current CLISYMTBL setting is less, you can lower the setting to its original value once the installation is finished.)
GBLPAGES	30000 available pages (For systems where you are performing a reinstallation, this number is the current value of GBLPAGES when the RMONSTOP command procedure or the RMU Monitor Stop command has been executed. Also, if .AIJ journaling is enabled, add 1,200 per database to the GBLPAGES value.)
GBLPAGFIL	50 available pages (Necessary only if the installation includes running the IVP.)
GBLSECTIONS	160 available sections (For systems where you are performing a reinstallation, this number is the current value of GBLSECTIONS when the RMONSTOP command procedure or the RMU Monitor Stop command has been executed.)
MAXBUF	1200 bytes
PQL_DENQLM	1000 locks
PROCSECTCNT	32 sections

The following sections show:

- How to check system parameter values.
- How to change parameter values with the OpenVMS AUTOGEN command procedure.
- How to change the values for dynamic system parameters.

1.7.7.1 Checking GBLPAGES and GBLSECTIONS Values

To install and run Oracle Rdb, you must set the correct values for the GBLPAGES and GBLSECTIONS system parameters. If you plan to enable global buffers, the values described in this section may have to be adjusted, depending on your system configuration. See the Oracle Rdb Guide to Database Performance and Tuning for more information. To see how many unused global pages and global sections your system has, enter the following commands:

```
$ WRITE SYS$OUTPUT F$GETSYI ("FREE_GBLPAGES")
8900
$ WRITE SYS$OUTPUT F$GETSYI ("FREE_GBLSECTS")
90
```

Section 1.7.7.3 describes the procedures for changing system parameter values.

1.7.7.2 Checking Other System Parameter Values

To check the values of your system parameters, enter the following command to invoke the OpenVMS System Generation utility (SYSGEN):

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>
```

At the SYSGEN prompt (SYSGEN>), enter the SHOW command to display the value of a system parameter. The values displayed should equal or exceed the value of each parameter listed in Table 1−4. The following command displays the value for the MAXBUF system parameter:

```
SYSGEN> SHOW MAXBUF
Parameter Name        Current    Default     Min.       Max.   Unit  Dynamic
--------------        -------    -------   -------    -------  ----  -------
MAXBUF                 16384       8192      4096      64000   Bytes    D
```

After you finish checking the parameters with the SHOW command, you can enter the EXIT command at the SYSGEN prompt to return to command−line level.

Section 1.7.7.3 describes the procedures for changing system parameter values.

1.7.7.3 Changing System Parameter Values with AUTOGEN

You use the AUTOGEN command procedure to change system parameters. The AUTOGEN command procedure automatically adjusts values for parameters that are associated with the ones you set manually. To change system parameters with AUTOGEN, edit the SYS$SYSTEM:MODPARAMS.DAT file.

To change a parameter value that is already in the SYS$SYSTEM:MODPARAMS.DAT file, delete the current value associated with that parameter and enter the new value. To add a new value, add a line to the MODPARAMS.DAT file. The line contains the name of the parameter and its value. For example:

```
MIN_MAXBUF = 2048
```

You can also modify incremental parameters in the MODPARAMS.DAT file. The following example increases the global page setting by 2000:

```
ADD_GBLPAGES = 2000
```

After you have made all your changes, run the AUTOGEN procedure to recalculate your system parameters. Enter the following command at the prompt:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT
```

AUTOGEN automatically adjusts some of the SYSGEN parameters based on the consumption of resources since the last reboot. If you do not want this automatic adjustment, include the NOFEEDBACK parameter at the end of the AUTOGEN command line. The AUTOGEN procedure performs an automatic system shutdown and reboots when it has finished. Rebooting your system activates the new parameter values. For more information about using AUTOGEN, see the OpenVMS system management documentation.

1.7.7.4 Setting Dynamic System Parameters

You can use SYSGEN to change the values for dynamic system parameters. The following example demonstrates this process for the CLISYMTBL system parameter. (After the installation is complete, you can reset CLISYMTBL to its previous setting or let it be reset automatically when you reboot your system.)

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE ACTIVE
SYSGEN> SET CLISYMTBL 250
SYSGEN> WRITE ACTIVE
SYSGEN> EXIT
```

Dynamic parameters changed with the SYSGEN WRITE ACTIVE command become active immediately without any need to reboot your system. In fact, rebooting returns dynamic system parameter values to their previous settings. Once you set values for dynamic parameters, you should complete the installation before rebooting the system. The values for other dynamic parameters, such as MAXBUF, must remain at the same level or later than the values specified in Table 1−4.

1.7.8 Back Up Your System Disk

At the beginning of the installation, the VMSINSTAL command procedure asks if you have backed up your system disk. Back up your system disk before installing any software on top of the operating system.

This precaution protects your system software. A system failure at a critical point in the installation procedure could leave unusable files. You also protect an existing version of the product, which may, if you request it, be deleted during the installation. Use the backup procedures that have been established at your site. For details on backing up your system disk, see the OpenVMS system management documentation.

1.7.9 Avoid Giving Users Access to Online Help

When the installation inserts the Oracle Rdb Help Modules into the OpenVMS Help Library, it must have sole access to the OpenVMS Help Library. If anyone uses the HELP command when the installation tries to insert the Oracle Rdb Help Modules, the installation stalls. You can prevent other users from using Help during the installation by either of the following methods:

- Running the installation when no one else is logged in.
- Limiting access to the help library SYS$HELP:HELPLIB.HLB to the SYSTEM account. Remember to note the original protection on the library, which you can determine with the following command:

```
$ DIR/PROTECTION SYS$HELP:HELPLIB.HLB
```

You can limit help library access with the following command:

```
$ SET PROTECTION = (S:RWED, O, G, W) SYS$HELP:HELPLIB.HLB
```

After the installation, return the protection on the help library to the original setting.

1.7.10 Prevent Interactive Users from Gaining Access to the System

If the installation fails for an indeterminable reason, install Oracle Rdb again, keeping all interactive users off the system during the installation procedure. You might also choose to keep interactive users off the system if you will be changing any system parameter values with the AUTOGEN command procedure. Use the REPLY command to inform users of the schedule for the installation. Prevent other users from logging in by issuing the SET LOGIN command:

```
$ REPLY/USER "Installation of Oracle Rdb starting in 5 minutes. Please log out."
$ SET LOGIN/INTERACTIVE=0
```

Both of these commands require the OPER privilege. If any batch or device jobs are running, you have two options:

- Wait until the last job finishes.
- Use the DELETE/ENTRY command to stop any job that is still running.

1.7.11 Time Required

The time required for the installation varies depending on the type of installation media, system configuration, and whether or not you need to reboot your system. The installation (including the running of the Installation Verification Procedure (IVP)) takes approximately 10 minutes on an HP rx6600 server.

Chapter 2
Installing Oracle Rdb

This chapter describes how to install Oracle Rdb. Table 2−1 summarizes the preparatory tasks described in Chapter 1.

Section 2.1 describes how to obtain copies of the release notes. Section 2.2 contains a step−by−step description of the installation procedure. Section 2.6 presents common installation errors and their solutions.

Table 2−1 Preinstallation Checklist

Task	For More Information ...
Confirm required software and disk space requirements.	See Section 1.2 and Section 1.4
Back up existing databases.	See Section 1.7.1
Resolve repository considerations.	See Section 1.7.3
Confirm adequate account privileges.	See Section 1.7.4
Confirm account password is unlocked.	See Section 1.7.5
Confirm adequate account quotas.	See Section 1.7.6
Confirm system parameter values.	See Section 1.7.7
Back up your system disk.	See Section 1.7.8
Disable access to online help.	See Section 1.7.9
Prevent access to the system.	See Section 1.7.10

2.1 Accessing the Online Release Notes

The Oracle Rdb installation procedure copies the latest release notes to the SYS$HELP directory. You can specify OPTIONS N when you invoke the VMSINSTAL command procedure to see the release notes before continuing the installation. The installation provides text, PostScript, and PDF formats of the release notes:

- The file specification for the text format is SYS$HELP:RDB072xx.RELEASE_NOTES.
- The file specification for the PostScript format is SYS$HELP:RDB072xx_RELEASE_NOTES.PS.
- The file specification for the PDF format is SYS$HELP:RDB072xx_RELEASE_NOTES.PDF.

Printed release notes are not included with the documentation set for Oracle Rdb. Review the release notes in case they contain any information about changes in the installation procedure.

2.2 Installation Procedure

The Oracle Rdb installation process consists of a series of questions and informational messages.

2.2.1 Invoking VMSINSTAL

To start the installation, invoke the VMSINSTAL command procedure from a privileged account, such as the SYSTEM account. The VMSINSTAL command procedure is in the SYS$UPDATE directory. You can use the following syntax to invoke VMSINSTAL:

```
@SYS$UPDATE:VMSINSTAL variant−name device−name
```

Alternatively, you can just type @SYS$UPDATE:VMSINSTAL at the system prompt. VMSINSTAL will prompt you for the variant name and device names. The rest of this section describes these parameters.

- variant−name
  The variant of Oracle Rdb you want to install. For example, enter the following:
    - *RDBV72530IM072* for OpenVMS I64
    - *RDBV72531AM072* for OpenVMS Alpha
- device−name
  The name of the device on which the media is mounted.
    - If the device is a disk drive, such as a CD−ROM reader, you also need to specify a directory. For CD−ROM distribution, the directory name is the same as the variant name. For example:

      ```
      DKA400:[RDBV72530IM072]
      ```
    - If the device is a magnetic tape drive, you need to specify only the device name. For example:

      ```
      MTA0:
      ```

2.2.2 Steps of the Installation Procedure

This section discusses the installation process itself, presenting all questions that appear during the installation. This section presumes that you entered the product name and device name on the VMSINSTAL command line. Refer to Appendix A for a sample installation procedure of Oracle Rdb.

Each question in the installation is marked with an asterisk (*) at the beginning of the line. Some questions show the default response in brackets, for example, [YES]. To use the default response, press the Return key.

1. Mounting the media
   Mount the distribution volume on the device you specified. VMSINSTAL confirms the variant you are installing.

   ```
   The following products will be processed:
     RDBV72010IM V7.2
           Beginning installation of RDB V7.2 at 17:03

   %VMSINSTAL−I−RESTORE, Restoring product save set A...
   ```
2. Reviewing the release notes
   The installation procedure automatically copies the release notes to the following:

- ♦ SYS$HELP:RDB072xx.RELEASE_NOTES
- ♦ SYS$HELP:RDB072xx_RELEASE_NOTES.PS
- ♦ SYS$HELP:RDB072xx_RELEASE_NOTES.PDF

```
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
 Copyright © 1995, 2013, Oracle Corporation. All Rights Reserved.
```

Note

It is useful to keep the release notes for previous versions of Oracle Rdb.

3. Printing the installation guide
   The installation procedure now asks if you want to print the installation guide, which it copies to SYS$HELP:RDB072_INSTALL_GUIDE.PS (and .PDF and .TXT).

```
The Rdb installation guide will be provided in SYS$HELP.

* Would you like to print the installation guide ? [NO]:
```

4. If you have CDD/Repository or DECdesign installed on your system, you will see the following note.

```
           IMPORTANT **** PLEASE NOTE ******

The RDB$CONVERT_CDD$DATABASE.COM procedure
will be provided in SYS$LIBRARY.  This command procedure
should be used to upgrade each CDD/Repository database and
DECdesign library on your system.

Please see the Oracle Rdb V7.2 release notes for more details.
```

5. Confirming the installation
   VMSINSTAL confirms the installation and asks if you want to continue.

```
        Installation procedure for: "Oracle Rdb V7.2-010"

    You are about to install a multiversion Oracle Rdb kit.

    Be sure you have read the section entitled
        "Preparing Your System and the Installing Account"
    in the installation guide before continuing with the installation.

* Do you want to proceed [YES]?

    Checking system requirements ...
```

6. Entering a UIC for the RDB$REMOTE72 account
   If this is the initial installation of Oracle Rdb, the procedure creates an account called RDB$REMOTE72. You must choose a unique user identification code (UIC) for this account, which the installation procedure uses when it creates the RDB$REMOTE72 account. The installation procedure prompts you to enter a UIC.

```
    *************************************************************

    This installation requires the creation of the RDB$REMOTE72
    account.  The installation procedure will not proceed until you
    enter a valid user identification code (UIC) for the RDB$REMOTE72
    account.  The UIC must be unique.  Format [ggg,mmm].

    **************************************************************

* Enter UIC to be used for RDB$REMOTE72 account: [123,475]
```

7. Creating the RDMAIJ72 account

   The installation procedure requires the creation of the RDMAIJ72 account. You must choose a unique user identification code (UIC) for this account, which the installation procedure uses when it creates the RDMAIJ72 account. The installation procedure prompts you to enter a UIC.

   ```
       *************************************************************

       This installation requires the creation of the RDMAIJ72
       account.  The installation procedure will not proceed until you
       enter a valid user identification code (UIC) for the RDMAIJ72
       account.  The UIC must be unique.  Format [ggg,mmm].

       **************************************************************

   * Enter UIC to be used for RDMAIJ72 account: [123,101]
   ```
8. Creating the RDMSTT72 account

   The installation procedure requires the creation of the RDMSTT72 account. You must choose a unique user identification code (UIC) for this account, which the installation procedure uses when it creates the RDMSTT72 account. The installation procedure prompts you to enter a UIC.

   ```
       *************************************************************

       This installation requires the creation of the RDMSTT72
       account.  The installation procedure will not proceed until you
       enter a valid user identification code (UIC) for the RDMSTT72
       account.  The UIC must be unique.  Format [ggg,mmm].

       **************************************************************

   * Enter UIC to be used for RDMSTT72 account: [123,102]
   ```
9. Choosing to run the Installation Verification Procedure (IVP)

   The Installation Verification Procedure (IVP) checks that Oracle Rdb is correctly installed. It creates a sample database and processes and runs sample programs against it. The installation asks if you want to run the IVP. Oracle Rdb recommends that you run the IVP.

   ```
   * Do you want to run the IVP after the installation [YES]? YES
   ```

   As part of the IVP, Oracle Rdb creates the PERSONNEL sample database in the directory specified by the RDM$DEMO logical name. You can also run the IVP independently at any time after Oracle Rdb is installed. See Section 3.11.
10. Choosing to purge files

    You have the option to purge files from previous versions of Oracle Rdb that are superseded by this installation. Purging is recommended; however, if you need to keep files from the previous version, enter NO in response to the question.

    ```
    * Do you want to purge files replaced by this installation [YES]?
    ```
11. Displaying informational messages

    At this point, the installation procedure displays a number of informational messages that report on the progress of the installation. There are no further questions. If the installation procedure has been successful up to this point, VMSINSTAL moves the new or modified files to their target directories, updates help files, and updates DCL tables, if necessary. If you asked for files to be purged, that work is done now. The following messages are displayed:

    ```
       There are no more questions.
    ```

```
    Beginning installation ...

    Installing under VMS XADH−T3Z   −  21−SEP−2004 17:04

%VMSINSTAL−I−RESTORE, Restoring product save set B ...
%VMSINSTAL−I−RESTORE, Restoring product save set C ...
%VMSINSTAL−I−RESTORE, Restoring product save set D ...
%VMSINSTAL−I−RESTORE, Restoring product save set E ...
   .
   .
   .
%VMSINSTAL−I−MOVEFILES, Files will now be moved to their target directories
. . .
```

12. Running the IVP
    If you chose to run the IVP, VMSINSTAL runs it now. When the IVP runs successfully, you see the following display:

    ```
    IVP completed for: Oracle Rdb V7.2−010
    ```
13. Completing the installation
    The following messages indicate that the entire installation procedure is complete:

    ```
     Installation of RDBV72010IM v7.2 completed at 11:09

       Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

       Creating installation data file: VMI$ROOT:[SYSUPD]RDBV72010IM072.VMI_DATA

     VMSINSTAL procedure done at 11:09
    ```

    Note that VMSINSTAL deletes or changes entries in the process symbol tables during the installation. Therefore, if you are going to continue using the system manager's account and you want to restore these symbols, you should log out and log in again.

2.3 Overview of Multiple−Version Support in Oracle Rdb

Oracle Rdb allows you to install and run multiple versions of Oracle Rdb software on a single OpenVMS system. This capability facilitates the process of upgrading to new versions of the software. You can now install the newest version of Oracle Rdb, use the RMU Convert command with the Nocommit qualifier to convert a database from an earlier version, and test your applications using this converted database. If you need to return to the previous version, use the RMU Convert command with the Rollback qualifier. Each database can be converted independently, but each database can be accessed by only one version of Oracle Rdb.

Multiple−version (multiversion) support is implemented by appending the Oracle Rdb software version number to Oracle Rdb file and image names. For example, the version of RDMSHR.EXE specific to Oracle Rdb Release 7.2 is named RDMSHR72.EXE; the image for interactive SQL is named SQL$72.EXE.

Because the multiversion files have the variant in the image name, installing the multiversion kit does not replace standard version files. Three files, RDBINTSHR.EXE, RDBSHR.EXE, and SQL$INT.EXE, are not varianted in either the standard version or the multiversion variant of Oracle Rdb. These files are guaranteed to be compatible with all versions of Oracle Rdb and are replaced only when a higher version of Oracle Rdb is installed on your system.

Note

Starting with Oracle Rdb Release 7.1, only multi−version installation is supported. Oracle Rdb Release 7.0 and earlier releases support a standard installation.

2.3.1 General Multiversion Support Considerations

Consider the following factors when you decide whether or not to install multiple versions of Oracle Rdb:

- By enabling multiversion support you can upgrade one database with its set of corresponding applications and test it before you upgrade all your databases and applications. You can also make a copy of the database and run parallel testing.
- Multiversion support requires disk space for each version of Oracle Rdb on the system disk. Furthermore, each version has its own demo programs, IVP files, help files, and message files that require additional space. As a rough guideline, double the block size for each version of Oracle Rdb on your system.
- Each version of Oracle Rdb requires a monitor process, RDMS_MONITOR or RDMS_MONITORnn (where nn is the version number).
- When multiple versions of Oracle Rdb are installed, each version of Oracle Rdb requires global pages to install shared images.

2.3.2 Oracle CDD /Repository Considerations

To install Oracle CDD/Repository Release 7.2 in a multiversion environment, take some or all of the following steps, depending on the combination of Oracle Rdb versions:

- To install Oracle CDD/Repository Release 7.2 in an Oracle Rdb Release 7.2 environment, use the RDB$SETVER command procedure, described in Section 2.4.1 to set up Oracle Rdb Release 7.2 as the active version during the installation. Failure to do this causes the Oracle CDD/Repository IVP procedure to fail.
- Oracle CDD/Repository Release 7.2 supports multiversioning. If you install Oracle CDD/Repository Release 7.2 and you also want to create a repository for a particular version of Oracle Rdb, (for example, Oracle Rdb Release 7.2), execute these steps:
  1. Use the RDB$SETVER command procedure to set up the Oracle Rdb Release 7.2 environment.

     ```
     $ @SYS$LIBRARY:RDB$SETVER 7.2 /SYSTEM
     ```
  2. Define the release 7.2 repository.

     ```
     $ REPOSITORY OPERATOR
     CDO> DEFINE REPOSITORY new_repository_name_72
     ```

     Because an Oracle CDD/Repository repository is an Oracle Rdb database, it has the on−disk structure of a particular version of Oracle Rdb. Thus, each repository can be used with only one version of Oracle Rdb. If you are using multiple versions of Oracle Rdb, you must have at least one repository for each version.

If you install Oracle CDD/Repository on a multiversion Oracle Rdb system and do not perform the installation from the SYSTEM account, you must use the RDB$SETVER command procedure to reset Oracle Rdb logical names. For example, if you have Oracle Rdb Release 7.2 set up as your environment and you install Oracle CDD/Repository from your process account, VMSINSTAL removes all process logical names. To redefine the Oracle Rdb Release 7.2 logical names, execute the following command:

```
$ @SYS$LIBRARY:RDB$SETVER 7.2
```

2.4 Accessing Multiple Versions of Oracle Rdb

This section describes how to:

- Change the default version of Oracle Rdb
- Set up process symbols to invoke images
- Determine which version or versions of Oracle Rdb are installed
- Link applications while running multiple versions
- Invoke LSE templates
- Access remote databases while running multiple versions
- Access online help for each version

2.4.1 Changing the Default Oracle Rdb Environment

After Oracle Rdb Release 7.2 is installed, you must run the RDB$SETVER.COM command procedure located in the SYS$LIBRARY directory. This procedure sets up logical names and symbols that establish a new Oracle Rdb environment.

If Oracle Rdb Release 7.2 is the only version of Oracle Rdb installed on the system, it is sufficient to run the following command in the system startup procedure:

```
$ @SYS$LIBRARY:RDB$SETVER 7.2 /SYSTEM
```

Individual users are not required to execute the RDB$SETVER command in their login procedures nor in the system−wide login procedure.

The RDB$SETVER command procedure accepts a parameter and a qualifier. The parameter specifies which version of Oracle Rdb you want to run (or reset, see Section 2.4.2). The qualifier specifies at which level the procedure defines logical names. For example:

```
$ @SYS$LIBRARY:RDB$SETVER 7.2 /SYSTEM
```

If you do not specify the parameter, the procedure prompts you for a version number:

```
$ @SYS$LIBRARY:RDB$SETVER
Enter MULTIVERSION version number (n.n) or S (for STANDARD): 7.2
Current PROCESS Oracle Rdb environment is version V7.2−0 (MULTIVERSION)
Current PROCESS SQL environment is version V7.2−0 (MULTIVERSION)
Current PROCESS Rdb/Dispatch environment is version V7.2−0 (MULTIVERSION)
```

The previous example sets the multiversion variant of Oracle Rdb Release 7.2 as the environment for the process that executed the RDB$SETVER command procedure. If you specify a version number for which no multiversion variant is available, the system verifies whether a standard version is available. If the standard version is available, the version is set to standard. If neither multiversion variant nor standard version is available, the system displays the following message:

```
$ @SYS$LIBRARY:RDB$SETVER S
%There is no Oracle Rdb STANDARD version on this system.
```

The RDB$SETVER command procedure can operate on the process, job, group, or system level. The default is /PROCESS. You can use RDB$SETVER.COM to establish the multiversion variant of Oracle Rdb as your default system environment by adding the @SYS$SYSTEM:RDB$SETVER.COM 7.2 command to SYSTARTUP_VMS.COM and specifying the /GROUP or /SYSTEM qualifier. You must have privileges to define group or system logical names to run RDB$SETVER.COM with the /GROUP or /SYSTEM qualifier.

The following list shows the logical names defined by the RDB$SETVER command procedure:

- RDB$DISPATCH_IDENT
- RDB$DISPATCH_VERSION_VARIANT
- RDBPRE
- RDBSERVER
- RDM$DEMO
- RDMS$VERSION_VARIANT
- RDMS$RMU_VARIANT
- RDBVMS$IDENT
- RDBVMS$IVP_DIR
- RDBVMS$LIB
- RDBVMS$OPTION
- RDBVMS$VARIANT
- RDBVMS$VERSION
- RDML
- RDMLRTL
- RDO
- RMUSHR
- RMUSTAT
- SQL$
- SQL$FUNCTIONS
- SQL$HELP_OLD
- SQL$IDENT
- SQL$MOD
- SQL$MSG
- SQL$PRE
- SQL$SAMPLE
- SQL$SHR
- SQL$USER
- SQL$VERSION_VARIANT
- SQLSRV$MOD

2.4.2 Setting Symbols with RDB$SETVER RESET

The RESET parameter of the RDB$SETVER command procedure sets symbols to invoke RMU and other Oracle Rdb images that correspond to the version number last set. This is important for RMU users who run the RDB$SETVER command procedure with the /GROUP or /SYSTEM qualifiers. In that case, other users' process−level symbols for RMU may not invoke the image corresponding to the version set by RDB$SETVER.COM. The procedure displays this informational message as a reminder:

```
$ @SYS$LIBRARY:RDB$SETVER 7.2 /SYSTEM
%You have changed the default Oracle Rdb Version at a level other
%than /PROCESS. The RMU symbol may have to be set by users
```

```
%using Oracle Rdb at this level. This can be done with the
%following DCL command: @SYS$LIBRARY:RDB$SETVER RESET
Current SYSTEM Oracle Rdb environment is version V7.2−0 (MULTIVERSION)
```

A user can determine if this incompatibility exists by examining the equivalence string for the logical name RDMS$VERSION_VARIANT and then executing the RMU Show Version command. The following example shows incompatibility between versions of Oracle Rdb and RMU:

```
$ SHOW LOGICAL RDMS$VERSION_VARIANT
   "RDMS$VERSION_VARIANT" = "72" (LNM$SYSTEM_TABLE)
$ RMU/SHOW VERSION
Executing RMU for Oracle Rdb V7.1−431
```

In the preceding example, a user can either change the version of RMU to Release 7.2, or change the version of Oracle Rdb to Release 7.1−431. Either way, a user must run the RDB$SETVER command procedure at the process level:

- Change the version of RMU to match the Oracle Rdb environment:

  ```
  $ @SYS$LIBRARY:RDB$SETVER RESET
  ```
- Change the Oracle Rdb environment to match the RMU version:

  ```
  $ @SYS$LIBRARY:RDB$SETVER S
  ```

In addition to setting up the appropriate symbol for RMU, RDB$SETVER RESET also creates symbols to invoke other Oracle Rdb interfaces:

```
$ SQL$ == "$SQL$"
$ SQL$PRE == "$SQL$PRE"
$ SQL$MOD == "$SQL$MOD"
$ RDML == "$RDML"
$ RDO == "$RDO"
$ RDBPRE == "$RDBPRE"
```

Note that image−invocation symbol definitions should not specify directories. For instance, you should not use either of the following symbol formats:

```
SQL == "RUN SYS$SYSTEM:SQL$"
SQL == "$SYS$SYSTEM:SQL$"
```

Both of these formats force the use of a specific image, and do not allow the use of variants.

2.4.3 Matching Environment and Database Versions

The RDB$SETVER.COM command procedure sets logical names and symbols for most Oracle Rdb images to point to variented file names. Thus, the symbol SQL$ points to SQL$72.EXE, and SQL$PRE points to SQL$PRE72.EXE, if you have set the version to 7.2. The following examples show how to determine your Oracle Rdb environment:

```
$ RMU/SHOW VERSION
Executing RMU for Oracle Rdb V7.2−00
$ RUN SQL$
SQL> ATTACH 'FILENAME PERSONNEL';
SQL> SHOW VERSIONS
```

```
Current version of SQL is: SQL V7.2−0
Underlying versions are:
    Database with filename PERSONNEL
        Oracle Rdb V7.2−0
        Rdb/Dispatch V7.2−0
SQL> DISCONNECT ALL;
SQL> EXIT;
```

To identify the version of Oracle Rdb associated with a database, use the RMU Show Version command, as follows:

```
$ RMU /SHOW VERSION MF_PERSONNEL
Executing RMU for Oracle Rdb V7.2
Database DUA0:[MFP]MF_PERSONNEL.RDB;1 requires version 7.0
```

The following example shows the error messages displayed if you try to attach to a database with the incorrect version of Oracle Rdb:

```
SQL> ATTACH 'FILENAME PERSONNEL';  ! This is a 7.1 database
%SQL−F−ERRATTDEC, Error attaching to database personnel
−RDB−F−WRONG_ODS, the on−disk structure of database filename is not supported
by version of facility being used
−RDMS−F−ROOTMAJVER, database format 71.0 is not compatible with software
version 72.0
SQL> SHOW VERSION
Current version of SQL is: SQL V7.2−0
```

2.4.4 Identifying Environment Versions with RDB$SHOVER

Layered and third−party products can determine which version or versions of Oracle Rdb are installed on their systems by using the RDB$SHOVER command procedure. Previously, these products usually read the version number from the header of one of the standard Oracle Rdb images, such as RDMSHRP. If you install the multiversion variant of Oracle Rdb Release 7.2, the old image names may not be available to determine the version number. The RDB$SHOVER.COM procedure (located in SYS$LIBRARY) allows four optional parameters. If you set P1 to VERSIONS, the process logical name RDBVMS$INSTALLED_VERSIONS is defined as a list of the Oracle Rdb versions. Each installed version has the following format:

```
 [*]AM.N[U]−cc
```

- An asterisk (*) denotes a varianted version.
- The A can be either a V for a released version or a T for a field test version.
- The M indicates the major version.
- The N indicates the minor version.
- The U indicates letter variants for mandatory update (MUP) releases.
- The cc indicates the count number.

The following example shows how to run the RDB$SHOVER command procedure interactively:

```
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS
   "RDBVMS$INSTALLED_VERSIONS" = "V7.1−0" (LNM$PROCESS_TABLE)
        = "*V7.2−0"
```

In this example, V7.1−0 indicates that Oracle Rdb Release 7.1 is installed; *V7.2−0 indicates that the multiversion variant of Oracle Rdb Release 7.2 is installed. The following example shows a command procedure you could use to determine which versions of Oracle Rdb are available:

```
$ X=0
$ LP:
$ Y=F$TRNLNM("RDBVMS$INSTALLED_VERSIONS",,X)
$ IF Y .EQS. "" THEN GOTO FINISH
$ SHOW SYMBOL Y
$ X=X+1
$ GOTO LP
$ FINISH:
```

If you set P1 to VERSIONS and P2 to a specific version, for example, 7.2, the logical names show only the information of the version indicated.

```
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS 7.2
   "RDBVMS$INSTALLED_VERSIONS" = V7.2−0 LNM$PROCESS_TABLE)
```

If you set P1 to VERSIONS and P2 to ALL, process logical names listing SQL and Rdb/Dispatch versions are also displayed.

```
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS ALL
   "RDBVMS$INSTALLED_VERSIONS" = "V7.1−0" (LNM$PROCESS_TABLE)
        = "*V7.2−0"
   "SQL$INSTALLED_VERSIONS" = "V7.1−0" (LNM$PROCESS_TABLE)
        = "*V7.2−0"
   "RDB$DISPATCH_INSTALLED_VERSIONS" = "V7.1−0" (LNM$PROCESS_TABLE)
        = "*V7.2−0"
```

To suppress the display of the logical names, set the last parameter to NOSHOW.

```
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS NOSHOW
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS ALL NOSHOW
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS 7.2 ALL NOSHOW
```

2.4.5 Linking Programs

The RDB$SETVER command procedure defines the logical name SQL$USER. The translation of SQL$USER depends on which version of Oracle Rdb you have selected with RDB$SETVER.COM. For example, if you have specified release 6.0, SQL$USER translates to the SQL user library SQL$USER60.OLB; if you have specified release 7.2, SQL$USER translates to SQL$USER72.OLB.

The RDB$SETVER command procedure does not define the logical name LNK$LIBRARY, which enables users to link embedded SQL programs without explicitly specifying an SQL library. By defining LNK$LIBRARY as SQL$USER, users can automatically link SQL programs to the version of the SQL library established by RDB$SETVER.COM. You can define LNK$LIBRARY as a system logical by using the following command:

```
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG LNK$LIBRARY SQL$USER
```

Section 3.2.3 provides additional information about LNK$LIBRARY and SQL$USER.

2.4.6 Using LSE Templates in SQL

The LSE (DEC Language−Sensitive Editor) templates allow users of interactive SQL and SQL module language to develop programs quickly and accurately.

Note

The LSE templates provided with Oracle Rdb provide support for SQL syntax only through release 4.2. The templates do not provide support for new and changed syntax after release 4.2.

With the multiversion variant of Oracle Rdb, LSE templates for each installed version of Oracle Rdb are available. After you have established a default Oracle Rdb environment using the RDB$SETVER command procedure, you must define a logical name to point to the appropriate LSE environment (.ENV) file. As you toggle between versions of Oracle Rdb, you must set the LSE environment accordingly.

LSE templates for the multiversion variant of Oracle Rdb Release 7.2 are located in SYS$LIBRARY:LSE$SQL72MV_ENVIRONMENT.ENV. To access SQL syntax, you must use one of the following methods:

- Use an LSE qualifier when invoking the LSEDIT editor. You must specify the complete device and directory name.

  ```
  $ LSEDIT /SYSTEM_ENVIRONMENT=SYS$LIBRARY:LSE$SQL72MV_ENVIRONMENT.ENV TEST.SQL
  $ LSEDIT /SYSTEM_ENVIRONMENT=SYS$LIBRARY:LSE$SQL72MV_ENVIRONMENT.ENV TEST.SQLMOD
  ```
- Define a process logical name first and invoke LSEDIT without a qualifier.

  ```
  $ DEFINE LSE$SYSTEM_ENVIRONMENT SYS$LIBRARY:LSE$SQL72MV_ENVIRONMENT.ENV
  $ LSEDIT TEST.SQL
  $ LSEDIT TEST.SQLMOD
  ```

2.4.7 Accessing Remote Databases

You can access a release 7.2 database on a remote node, even if your node is currently running an earlier version.

2.4.7.1 Using DECnet Transport

The LOGIN.COM command procedure for the RDB$REMOTE72 account defines the appropriate RDBSERVER and RDB$SHARE images to run. Users must specify the RDB$REMOTE72 account when they access a remote database. For example, to access the PERSONNEL database on node RAILS, enter the following command:

```
SQL> ATTACH 'FILENAME RAILS"RDB$REMOTE72 password"::DISK1:[DBASES]PERSONNEL';
```

To avoid displaying the password on the terminal screen, you can define proxies for appropriate users. See the Oracle Rdb7 Guide to SQL Programming for information about using proxies for remote access.

If you use a proxy account instead of using the RDB$REMOTE72 account, you must add the following lines to the account's LOGIN.COM file:

```
$ DEFINE RDBSERVER SYS$COMMON:RDBSERVER72.EXE
$ DEFINE RDMS$VERSION_VARIANT 72
```

2.4.7.2 Using TCP/IP Transport

You can define your own UCX service to access an earlier version of a database when using the TCP/IP transport. You must define this service to have a user name that is set up for the earlier version of Oracle Rdb. For example, to access a V6.1 database you can create a UCX service called rdbsrv61 that uses the user name rdb$remote61. Then add the following line to your client configuration file to use the UCX service:

```
SQL_ALTERNATE_SERVICE_NAME rdbsrv61
```

For more details on how to set up UCX services, see Section 4.1.2. For more information about configuration files, see Section 4.3.1 and Section 4.3.2.

In this example, if you choose to use a different user name than rdb$remote61 to access a V6.1 database, the LOGIN.COM file of that user must contain the following lines:

```
$ DEFINE RDBSERVER SYS$COMMON:RDBSERVER61.EXE
$ DEFINE RDMS$VERSION_VARIANT 61
```

2.4.8 Accessing Online Help

When you install Oracle Rdb, users can access online help from the command line for each installed version of Oracle Rdb. The following is a list of varianted online help topics:

- ORACLE_RDB
- RDBPRE
- RDML
- RDO
- RMU
- SQL
- SQLMOD
- SQLPRE
- SQL_SERVICES

To access help on any multiversion variant installed on the system, type HELP and the topic name with a two−digit suffix representing the version. For example, to access the release 7.2 help on any of the varianted topics in the previous list, type HELP and the topic name with a "72" suffix:

```
$ HELP SQL72
```

You can invoke help on SQL statements while you are in interactive SQL by typing the following:

```
SQL> HELP
```

2.5 How Applications Access Multiple Versions of Oracle Rdb

The following images are installed in SYS$COMMON:[SYSLIB] by VMSINSTAL:

- RDB$SHARE72.EXE
- RDBSHR.EXE

Many layered products and third−party products call RDBSHR.EXE at image activation time. With the multiversion variant of Oracle Rdb, more than one version of Oracle Rdb is available to an application. The version required depends on the parameter set by RDB$SETVER.COM.

Applications still call RDBSHR.EXE but RDBSHR.EXE checks only what version the application wants to use by examining the logical name RDMS$VERSION_VARIANT. If RDMS$VERSION_VARIANT is not defined, RDBSHR.EXE calls RDB$SHARE.EXE, which contains the current released version code. For example, if RDMS$VERSION_VARIANT translates to 71, RDBSHR.EXE calls RDB$SHARE71.EXE, which contains the release 7.1 code.

2.6 Errors That Cause the Installation or IVP to Fail

If errors occur during the installation itself or when the IVP is running, VMSINSTAL displays failure messages. If the installation fails, you see the following message:

```
%VMSINSTAL−E−INSFAIL, The installation of RDB V7.2 has failed.
```

If the IVP fails, you see these messages:

```
The RDB V7.2 Installation Verification Procedure failed.

%VMSINSTAL−E−IVPFAIL, The IVP for RDB V7.2 has failed.
```

Errors can occur during the installation if any one of the following conditions exists:

- Incorrect version of OpenVMS
- Incorrect version of Oracle Rdb already installed
  If you have a version prior to release 4.0 already installed on your system, this multiversion installation will fail.
- Insufficient privileges
  The account you use to install Oracle Rdb must have the SETPRV privilege. See Section 1.7.4.
- Insufficient disk space on system disk
  If the system disk does not have enough blocks available to install Oracle Rdb, purge or delete unnecessary files according to the policies of your site. When you have enough disk space, you are ready to restart the installation procedure.
  See Section 1.4 for disk space requirements.
- Insufficient system parameter values
  You must have the necessary minimum settings for system parameters. See Section 1.7.7.
- Insufficient quotas for successful installation
  You must have the necessary minimum account quotas set. See Section 3.6.
- OpenVMS Help Library currently in use
- RMONSTART72.COM procedure found in SYS$SPECIFIC:[SYS$STARTUP]
  The IVP will fail if it executes an old version of the RMONSTART72.COM procedure that may have been inadvertently written in the SYS$SPECIFIC:[SYS$STARTUP] directory. Although the installation creates the file in SYS$COMMON:[SYS$STARTUP], you can inadvertently write it to SYS$SPECIFIC after editing the file.
  The installation procedure checks for RMONSTART*.COM in SYS$SPECIFIC:[SYS$STARTUP]. If it finds any files, it asks if you want to abort the installation. To prevent problems when you run the IVP, you should abort the installation, remove any RMONSTART*.COM files from SYS$SPECIFIC:[SYS$STARTUP], and run the installation again.

2.7 Japanese Rdb Kit Included with the Oracle Rdb Release 7.2 Media

The Oracle Rdb Release 7.2 media also contains the Japanese Rdb kits. After installing Oracle Rdb, you can use the VMSINSTAL command procedure to install the Japanese Rdb kit.

The save set names for the Japanese Rdb kits are:

- JRDBV72530AM072 for OpenVMS Alpha operating systems
- JRDBV72531AM072 for OpenVMS Alpha operating systems with EV56 and later processors
- JRDBV72530IM072 for OpenVMS I64 operating systems

Chapter 3
After Installing Oracle Rdb

This chapter describes required and optional tasks after installing Oracle Rdb. The following list summarizes those tasks.

Table 3−1 Postinstallation Checklist

Task	For More Information ...
Reset logins and help file protection.	See Section 3.1
Edit system startup and shutdown files.	See Section 3.2.1 and Section 3.2.2
Define LNK$LIBRARY and SQL$USER logical names (optional).	See Section 3.2.3
Modify system parameters.	See Section 3.3
Reboot the system (optional).	See Section 3.4
Activate Oracle Rdb for cluster members.	See Section 3.5
Modify user account privileges and quotas.	See Section 3.6
Convert existing databases.	See Section 3.7
Enable SQL SET LANGUAGE (optional).	See Section 3.8.1
Enable Oracle Trace support (optional).	See Section 3.8.2
Enable RDB$REMOTE72 account (optional).	See Section 3.8.3
Install images as resident on OpenVMS (optional).	See Section 3.9
Start Oracle CDD/Repository (optional).	See Section 3.10
Run the Installation Verification Procedure (IVP) (optional).	See Section 3.11
Reset read−only storage areas.	See Section 3.12
Delete previous versions of Oracle Rdb.	See Section 3.13

3.1 Returning the System to Original Settings

If you have set interactive logins to 0 or changed the protection on the help library, you must reverse these actions.

- To restore interactive logins, enter the following command: *$ SET LOGIN/INTERACTIVE=value*
- To change the protection on the help library, enter the following commands:

  ```
  $ SET DEFAULT SYS$HELP
  $ SET PROTECTION=(S:RWED,O:RWED,G:RWED,W:RE) HELPLIB.HLB
  ```
- If the system parameter CLISYMTBL was less than 512 before the installation, you can now set it to the original setting. See Section 1.7.7.4 for more information.

3.2 Starting and Shutting Down Oracle Rdb

You must edit system startup and shutdown files to provide for automatic startup and shutdown of Oracle Rdb when your system is rebooted.

3.2.1 Editing the System Startup File

Edit SYS$STARTUP:SYSTARTUP_VMS.COM and add the command that starts Oracle Rdb.

You must position this new command line after the line that invokes the network startup command procedure. The following example shows the network startup command line followed by the startup command line for Oracle Rdb:

```
$ @SYS$MANAGER:STARTNET.COM
     .
     .
     .
$ @SYS$STARTUP:RMONSTART72.COM
```

Because you have installed a multiversion variant of Oracle Rdb, you must include a command line that starts each version of Oracle Rdb running on your system. In the following example, RMONSTART.COM starts a previously installed version of Oracle Rdb, and RMONSTART72.COM starts the multiversion variant of Oracle Rdb Release 7.2.

```
$ @SYS$MANAGER:STARTNET.COM
     .
     .
     .
$ @SYS$STARTUP:RMONSTART.COM
$ @SYS$STARTUP:RMONSTART72.COM
```

You should also consider editing the system startup file to run the RDB$SETVER.COM procedure to establish a default Oracle Rdb environment. See Section 2.4.1 for more information.

Note

The STARTUP commands of the SYSMAN utility provide an alternative to editing system startup files to invoke RMONSTART72.COM. See the OpenVMS system management documentation for more information.

3.2.2 Editing the System Shutdown File

Add the following command line to the system shutdown file, SYS$MANAGER:SYSHUTDWN.COM, to shut down Oracle Rdb when the system is shut down:

```
$ @SYS$MANAGER:RMONSTOP72.COM
```

You must include the command line to shut down each version of Oracle Rdb running on your system, for example:

```
$ @SYS$MANAGER:RMONSTOP.COM
$ @SYS$MANAGER:RMONSTOP72.COM
```

To invoke the RMONSTOP72.COM command procedure, you need the user privilege SETPRV or the privileges CMKRNL, SYSNAM, and WORLD. The RMONSTOP72.COM file includes the RMU Monitor Stop command with the Wait qualifier to stop the Oracle Rdb monitor.

3.2.3 Defining LNK$LIBRARY and SQL$USER to Ease Program Linking

Note

If you have installed any multiversion variant or standard version of Oracle Rdb and have run RDB$SETVER.COM, SQL$USER is automatically defined to point to the correct version of the SQL user library. See Section 2.4.5

If you define the logical name LNK$LIBRARY as the SQL user library, users will not have to explicitly specify that library each time they link their embedded SQL programs. To define LNK$LIBRARY as a system−wide logical name, issue this command:

```
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG LNK$LIBRARY SQL$USER
```

To make sure LNK$LIBRARY is defined each time the system starts up, add the previous command to your system startup procedure. If you do not define SQL$USER and LNK$LIBRARY to specify the SQL user library, users must explicitly name it when they link programs with embedded SQL statements. For example:

```
$ LINK MY_PROG, SYS$LIBRARY:SQL$USER72.EXE/LIBRARY
```

See the OpenVMS documentation set for more information about the LINK command.

3.3 Modifying System Parameters

Depending on the other layered products installed on your system, you may have to adjust system parameters to improve Oracle Rdb performance. The values appropriate for your system might differ substantially from those values specified in Section 1.7.7. For instance, you might have to add the values you estimate for Oracle Rdb applications to the values calculated for other layered products.

Table 1−–4 lists the minimum system parameter values required to install Oracle Rdb. These values may result in satisfactory performance. However, if you are using these values and still have Oracle Rdb performance problems, see the Oracle Rdb7 Guide to Database Performance and Tuning.

Optimizing the values for the GBLPAGFIL and GBLPAGES parameters is especially important if any database uses global buffers. Using global buffers increases performance in some applications because I/O is reduced and memory is better used. Refer to the Oracle Rdb7 Guide to Database Performance and Tuning for more information on how the GBLPAGES parameter affects performance when global buffers are enabled. GBLPAGFIL defines the maximum number of pages allowed for each global section. Determining a value for GBLPAGFIL depends on many factors, including the number of databases, the number of run units, the number and size of each global buffer, and the overhead.

An example of how you might calculate the requirement for GBLPAGFIL for one database is:

```
(# of database global buffers * size of each global buffer) * 2
```

If you use more than one database at a time, calculate the requirement for each database. If you change the GBLPAGFIL parameter, you must reboot your system.

3.4 Rebooting the System

You can reboot your system after you have installed Oracle Rdb, edited the system startup and shutdown files, and set the system parameters (if necessary). A system reboot performs the following operations:

- Verifies that Oracle Rdb is ready for use (that is, if you have added RMONSTART72.COM to the system startup file)
- Ensures that the edits to the system startup command file are correct
- Establishes any new parameter settings

Note that rebooting is optional.

3.5 Enabling Oracle Rdb on Other Cluster Nodes

If the system on which you installed Oracle Rdb is a member of a cluster environment, take the following steps to make Oracle Rdb available to other cluster members:

1. Edit the system startup and shutdown files of the cluster members on which you want to run Oracle Rdb so they invoke the Oracle Rdb startup and shutdown procedures. (You may omit this step if you have already made these changes in a command file that is invoked for all cluster systems.)
2. Reset the DCL tables on each node of the cluster.

   ```
   $ RUN SYS$SYSTEM:SYSMAN
   SYSMAN> SET ENVIRONMENT/CLUSTER
   SYSMAN> DO INSTALL REPLACE
   SYS$COMMON:[SYSLIB]DCLTABLES.EXE/OPEN/HEADER/SHARE
   ```

   You must log out and log in again on each node for the new DCL tables to take effect. If you do not, existing processes will not recognize the correct version of Oracle RMU.
3. Run the Oracle Rdb startup command procedure, RMONSTART72.COM, on each node in the cluster. The installation procedure ran this startup procedure on the processors on which you installed Oracle Rdb, so it is not necessary to rerun it from that CPU node. See Section 3.5.1.
4. After running the startup file, run the IVP on all other nodes to verify that Oracle Rdb is accessible from each node. See Section 3.5.1.
5. Run one of the following command files (depending on whether you have a DECnet Phase IV or a DECnet−Plus environment):
   - For DECnet Phase IV environments, run SYS$MANAGER:RDBSERVER_NCP.COM. Note that this command procedure is called and runs from RMONSTART72.COM. See Section 3.5.2 for more information on RDBSERVER_NCP.COM.
   - For DECnet−Plus environments, run SYS$MANAGER:RDBSERVER_NCL.COM. This command procedure is called and runs from RMONSTART72.COM. See Section 3.5.3 for more information on RDBSERVER_NCL.COM.

3.5.1 Using SYSMAN to Run Startup Procedures and Run the IVP on Each Node

You can use SYSMAN to run the Oracle Rdb startup procedure and the IVP on each node of your cluster environment. Enter the following commands to perform these operations on all nodes of a cluster:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT /CLUSTER /USERNAME=SYSTEM
Remote Password: <supply SYSTEM password here>
SYSMAN> DO @SYS$STARTUP:RMONSTART72
SYSMAN> DO @SYS$TEST:RDB$IVP72
SYSMAN> EXIT
```

If you want to perform these operations on only certain nodes of a cluster, substitute the /NODE qualifier for the /CLUSTER qualifier in the preceding example, and provide the names of the nodes on which you want to perform the operations (/NODE=(NODE1,NODE2)).

3.5.2 Executing RDBSERVER_NCP.COM in a DECnet Phase IV Environment

If you have DECnet−Plus installed on your system, read Section 3.5.3.

Log in to each node and run the RDBSERVER_NCP.COM procedure to insert the RDBSERVER object into the permanent DECnet object database of that node. You must execute the procedure once per cluster node. You do not have to execute it on the node from which the installation took place, because the installation procedure that executes on that node performs the RDBSERVER insertion.

The RDBSERVER_NCP.COM procedure configures the DECnet RDBSERVER object through the NCP command interface. It assumes that the network permanent database file is a cluster one. If there is any error configuring the RDBSERVER object, the system displays instructions to help you configure the RDBSERVER object manually.

Note

> ***RDBSERVER_NCP.COM is also called by SQL$STARTUP.COM, which is called by RMONSTART72.COM. If you execute RMONSTART72.COM interactively on other nodes after the installation, you do not have to invoke RDBSERVER_NCP.COM.***

3.5.3 Executing RDBSERVER_NCL.COM in a DECnet−Plus Environment

If you have DECnet Phase IV installed on your system, read Section 3.5.2. Log in to each node and run the RDBSERVER_NCL.COM procedure to configure the RDBSERVER object with the DECnet−Plus database. You must execute RDBSERVER_NCL.COM once per cluster node. You do not have to execute the RDBSERVER_NCL.COM procedure on the node from which the installation took place. RMONSTART72.COM calls RDBSERVER_NCL.COM to configure RDBSERVER.

If the installation procedure is on cluster node NODE1 and if the cluster system also includes nodes NODE2 and NODE3, you must log in to nodes NODE2 and NODE3 and enter the following:

```
$ SET DEFAULT SYS$STARTUP
$ @RDBSERVER_NCL
```

Note

> ***RDBSERVER_NCL.COM is also called by SQL$STARTUP.COM, which is called by RMONSTART72.COM. If you execute RMONSTART72.COM on other nodes after the installation, you do not have to invoke RDBSERVER_NCL.COM.***

The following error may occur when you run the RDBSERVER_NCL.COM procedure:

```
Node 0 Session Control Application RDBSERVER

command failed due to:
 access denied
```

The error may also occur when you run the RMONSTART72.COM procedure or when you install the product.

If you see this error, check the DECnet−Plus documentation for information on how to correct it. After you have corrected the error, rerun RDBSERVER_NCL.COM to configure the RDBSERVER network object.

3.6 Minimum User Account Privileges and Quotas

To work with Oracle Rdb, Oracle suggests that user accounts should have these minimum quotas:

Table 3−2 Suggested Minimum Process Quotas

Quota	Suggested Minimum
FILLM	25 more than the total number of database storage areas, snapshot storage areas, and after image journals
BYTLM	The larger of 1,048,576 or 512 times the sum of the following: ♦ 1024 (for Sort work, AIJ and RUJ IO operations) ♦ Database asynch batch write buffer count times the database buffer size ♦ Database asynch prefetch buffer count times the database buffer size ♦ Number of database storage areas, snapshot storage areas and AIJ files
WSQUOTA, WSEXTENT	Large enough to avoid excessive process page faulting
DIOLM	8. Larger values combined with high performance storage subsystems and large asynchronous IO counts can allow increased throughput.
ENQLM	Oracle suggests a value of 32767 in the UAF account entry to permit a virtually unlimited number of database−related locks being held
BIOLM	16
ASTLM	The larger of 100 or the sum of the following: ♦ 5 (for Sort work, AIJ and RUJ IO operations) ♦ Database asynch batch write buffer count times the database buffer size ♦ Database asynch prefetch buffer count times the database buffer size
PGFLQUO	Large enough to contain the process's program use of buffers and code along with Rdb's use of buffers and code. The consumption of a process's virtual private pages (and, by definition, possible use of page file space) is related to numbers of database buffers, numbers of storage areas, tables, locks, query complexity and so on and is very application and configuration dependent. Oracle suggests that a value of 1000000 be considered as adequate for many database applications.
TQELM	100

You use AUTHORIZE to verify and change user accounts. You must have system privileges to use AUTHORIZE. At the AUTHORIZE prompt (UAF>), enter the SHOW command with an account name to check that particular account. For example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW SMITH
```

To change quotas and privileges, use the MODIFY command:

```
MODIFY account−name /quota−name=NNN /PRIVILEGE=(priv−name) /DEFPRIV=(priv−name)
```

The following example changes the FILLM quota for the SMITH account, and gives it the TMPMBX and NETMBX privileges:

```
UAF> MODIFY SMITH /FILLM=300 −
_UAF> /PRIVILEGE=(TMPMBX,NETMBX) /DEFPRIV=(TMPMBX,NETMBX)
```

Users must log out and log in again for changes made in AUTHORIZE to take effect. For more information on modifying account quotas, see the description of the OpenVMS Authorize utility in the OpenVMS system management documentation.

3.7 Converting Existing Databases

Users must use Oracle RMU to convert existing Oracle Rdb databases to a format compatible with Oracle Rdb Release 7.2 software. Existing databases include those associated with Oracle CDD/Repository, Oracle Trace and other layered products. You can directly convert release 7.0 and later databases using the RMU Convert command. See Section 1.7.1 for additional information.

Users converting databases with the RMU Convert command must be sure their processes access the DCLTABLES shared image replaced by the Oracle Rdb installation procedure:

1. All cluster nodes must have replaced the image (see Section 3.5.1).
2. Users must log out and log in again.

The RMU Convert command accepts the database file name you enter, updates all metadata, and creates new metadata for Oracle Rdb Release 7.2. You can use a list of specific database names that may include wildcards. You can also specify a repository path name using the Path qualifier. However, wildcards are not allowed for repository path names. To convert a database to a format compatible with Oracle Rdb Release 7.2, perform the following steps:

1. Back up the pre−release 7.2 Oracle Rdb database.
2. Enter the RMU Convert command:

```
$ RMU/CONVERT <db−filename>
```

By default, RMU commits the conversion unless you specify Nocommit on the command line. The Nocommit qualifier lets you postpone either committing the conversion or rolling it back. If you have specified Nocommit, the RMU Convert command leaves two versions of the metadata in your database. You can use the database with the newer version of Oracle Rdb, or you can also use release 4.0 through release 7.1 to access databases that have not been converted.

The multiversion feature of Oracle Rdb enables you to test applications using the latest version of Oracle Rdb, while continuing to use databases with the previous version of the software. However, you will not be able to perform data definition language (DDL) operations on that database until after you commit the conversion. If you specify the Commit qualifier, RMU will create a new version of your metadata, and delete the old version.

Note

Once you have committed the conversion of a database file, you can no longer use that database file with a previous version of Oracle Rdb.

You can also specify the Rollback qualifier with the RMU Convert command. The Rollback qualifier specifies that the database should be rolled back to the old version. The following is an example of using the Rollback qualifier after specifying Nocommit:

```
$ RMU/CONVERT/NOCOMMIT PERSONNEL
    .
    .
    .
$ RMU/CONVERT/ROLLBACK PERSONNEL
```

```
Are you satisfied with your backup of DISK2:[USER]PERSONNEL.RDB;1 [N]? Y
After−image journaling will be disabled if the RMU/CONVERT of
DISK2:[USER]PERSONNEL.RDB;1 continues.  Do you wish to proceed [N]? Y
%RMU−I−LOGCONVRT, database root converted to current structure level
%RMU−I−CVTROLSUC, CONVERT rolled−back for DISK2:[USER]PERSONNEL.RDB;1 to
version V7.0
%RMU−I−LOGCREAIJ, created after−image journal file DISK2:[USER]PERSJL.AIJ;4
```

Users trying to access unconverted databases with release 7.2 software receive the following fatal error messages:

```
%RDB−F−WRONG_ODS, the on−disk structure of database filename is not
supported by version of facility being used
−RDMS−F−ROOTMAJVER, database format 70.0 is not compatible with software
version 72
%RDO−F−BACKCONV, Please backup your database and use the RMU/CONVERT
command.
```

The RMU Convert command copies all metadata in the system tables. Therefore, the time needed to convert a database depends upon the size of the system tables. If you have made many metadata changes, your system tables may be larger than if your metadata has been stable. If your database has very large system tables, it might be more efficient to use the SQL EXPORT and IMPORT statements to convert the database. An EXPORT/IMPORT operation involves only the latest version of the metadata; it does not make an exact copy of the system tables.
The RMU Convert command displays a question about the backup of your database. For example:

```
Are you satisfied with your backup of DISK$1:[RDB.DB]SHIPPING.RDB;48?
```

The RMU Convert command can disable after−image journaling during the conversion. If the database to be converted has after−image journaling enabled, RMU prompts you to determine if you want after−image journaling disabled so that the conversion can continue. If you reply N (for NO), the RMU Convert operation does not proceed and RMU returns you to command−line level.

```
$ RMU/CONVERT/NOCOMMIT PERSONNEL
Are you satisfied with your backup of DISK2:[USER]PERSONNEL.RDB;1 [N]? YES
After−image journaling will be disabled if the RMU/CONVERT of
DISK2:[USER]PERSONNEL.RDB;1 continues.  Do you wish to proceed [N]? NO
$
```

If you reply YES, RMU disables after−image journaling, converts the database, and then reenables after−image journaling with an .AIJ file of the same name and next higher version number:

```
$ RMU/CONVERT/NOCOMMIT PERSONNEL
Are you satisfied with your backup of DISK2:[USER]PERSONNEL.RDB;1 [N]? YES
After−image journaling will be disabled if the RMU/CONVERT of DISK2:[USER]
PERSONNEL.RDB;1 continues.  Do you wish to proceed [N]? YES
%RMU−I−LOGCONVRT, database root converted to current structure level
```

```
%RMU−S−CVTDBSUC, database DISK2:[USER]PERSONNEL.RDB;1 successfully
converted from version V7.0 to V7.2
%RMU−I−LOGCREAIJ, created after−image journal file
DISK2:[USER]PERSJL.AIJ;3
```

If you have already disabled after−image journaling, this prompt does not appear. If an error occurs when you use the RMU Convert command, restore the database (using the RMU Restore command) from the backup file created before the installation (see Section 1.7.1). If the system fails during the initial convert operation, reenter the RMU Convert command. If the RDB$SYSTEM storage area is read−only, RMU Convert automatically converts the RDB$SYSTEM storage area to read/write. If you want this storage area to be read−only, execute the following statement:

```
SQL> ALTER DATABASE FILENAME MY_DB
cont> ALTER STORAGE AREA RDB$SYSTEM READ ONLY;
```

Note

RMU Convert and SQL IMPORT operations create an RMU access control list (ACL) on earlier Oracle Rdb databases. The conversion bases the ACL on information from the Oracle Rdb internal database ACL and from any previously existing root file ACL on the original database. The RMU ACL created by the conversion attempts to provide a measure of backward compatibility for RMU command access, but it is unlikely that the resulting root file ACL will meet all the needs of database users. Modify the root file ACL with the RMU Set Privilege command to give access to users for needed RMU commands. See the Oracle Rdb for OpenVMS Oracle RMU Reference Manual for a description of the RMU Set Privilege command and the RMU Show Privilege command.

3. Backup the converted database immediately. The conversion operation creates a database that is different from the original. The .AIJ file corresponds to the newly converted database. If you need to perform an RMU Restore operation, you will need to apply the .AIJ file against the backup of the new database. For more information about RMU Convert, see the Oracle Rdb for OpenVMS Oracle RMU Reference Manual.

3.8 Tailoring Your System

This section provides information about special system arrangements and cleanup procedures that you can perform after installing Oracle Rdb.

3.8.1 Defining SYS$LANGUAGES

To allow you to use Oracle Rdb in the language or languages of your choice, define SYS$LANGUAGES as a list of all languages that you want. For example, if you want to be able to use English, Japanese, and French, define SYS$LANGUAGES as follows:

```
$ DEFINE SYS$LANGUAGES ENGLISH, JAPANESE, FRENCH
```

After defining SYS$LANGUAGES, run the following command procedure:

```
$ @SYS$STARTUP:LIB$DT_STARTUP.COM
```

Then you can use the SQL SET LANGUAGE statement to specify one of the languages defined by SYS$LANGUAGES. Refer to the Oracle Rdb7 SQL Reference Manual for more information on the LANGUAGE clause of the SQL SET statement and the SYS$LANGUAGES logical name.

3.8.2 Setting Up Oracle Trace

If you have Oracle Trace for OpenVMS installed on your system, you must manually restart Oracle Trace by running the EPC$STARTUP procedure. Enter the following command:

```
$ @SYS$STARTUP:EPC$STARTUP
```

The installation procedure inserts the Oracle Rdb facility definition into a library file called EPC$FACILITY.TLB. To be able to collect Oracle Rdb event data using Oracle Trace, you must move this facility definition into the Oracle Trace administration database. Perform the following steps:

1. Extract the definition from the facility library to a file (in this case, RDBVMS.EPC$DEF).

   ```
   $ LIBRARY /TEXT /EXTRACT=RDBVMSV7.2 /OUT=RDBVMS.EPC$DEF -
   _$ SYS$SHARE:EPC$FACILITY.TLB
   ```
2. Insert the facility definition into the Oracle Trace administration database.

   ```
   $ COLLECT INSERT DEFINITION RDBVMS.EPC$DEF /REPLACE
   ```

Note that the process executing the INSERT DEFINITION command must use the version of Oracle Rdb that matches the version used to create the Oracle Trace administration database or the INSERT DEFINITION command will fail.

The Oracle Rdb installation procedure may display an Oracle Trace error message if no Oracle Rdb monitor is running during the installation. This will be the case when you have stopped the RDMS_MONITOR process. The error message is informational and does not affect the installation. The message states that you must start the Oracle Rdb monitor before placing the facility definition in the Oracle Trace administration database.

3.8.3 Using the RDB$REMOTE72 Account for Remote Access

The Oracle Rdb installation creates the RDB$REMOTE72 account specifically for remote access. This account can be used by any program accessing any remote database. Programs that execute on remote nodes and access Oracle Rdb databases on your node through DECnet or TCP/IP can log in to your system through the RDB$REMOTE72 account.

3.8.3.1 DECnet and the RDBSERVER Object

For DECnet, the Oracle Rdb Release 7.2 installation procedure defines RDB$REMOTE72 as the default account for the RDBSERVER object. This definition supersedes any previous assignment you may have made for the RDBSERVER object.

The RDB$REMOTE72 account includes a password assigned by the system during the installation procedure. The password provided is used for the RDB$REMOTE72 account and in the DECnet object database on your node. This means that the RDB$REMOTE72 password and the password assigned to the RDBSERVER object will be the same. However, in a cluster environment, the installation procedure assigns the same password to the RDB$REMOTE72 account and the RDBSERVER object only on the node from which the installation took place. Be sure to make the proper assignments on each node that shares the common root directory.

Programs that execute on remote nodes and access an Oracle Rdb database on your node through DECnet can access your system through the RDB$REMOTE72 account, as long as the remote node allows RDB$REMOTE72 to access it. For example, to access an Oracle Rdb database on node TRIXIE from node NODE1, define a logical name for the remote file specification on node NODE1, enter SQL, and invoke the database:

```
$ ! On node NODE1:
$ DEFINE MYDB "TRIXIE::WORK$:[USER.DBS]PERSONNEL"
$ !                  ^
$ !                  |
$ ! Note there is no need for an access control string.
$ !
$ SQL
SQL> ATTACH 'FILENAME MYDB';
```

Because RDB$REMOTE72 is defined as the account used by the RDBSERVER object on node TRIXIE, it is not necessary (unless you specifically want the server to run under a different account) to include an access control string.

The RDB$REMOTE72 account is assigned the proper process quotas and privileges to work with Oracle Rdb. Some users have encountered problems with remote database access because they rely on the default DECnet account, which commonly does not have sufficient process quotas.

Note

> ***If the existing RDB$REMOTE72 account has the DISUSER flag set, then accessing the database through the RDB$REMOTE72 account will fail. The DISUSER flag disables the RDB$REMOTE72 account.***

The RDB$REMOTE72 account is a restricted account. It does not require a SYS$MANAGER:SYLOGIN.COM procedure. However, if you encounter any errors with the use of the RDB$REMOTE72 account, check that the SYS$SYLOGIN logical name (if defined) points to a working SYLOGIN.COM procedure.

RDB$REMOTE72 does require a login procedure. The login procedure for RDB$REMOTE72 is RDB$REMOTE_LOGIN72.COM; it resides in SYS$COMMON:[SYSEXE]. This login procedure includes security checks that ensure the user is running the RDBSERVER object (DECnet object number 35). If you want product−specific files to be run during the RDB$REMOTE72 account login step, you must edit the RDB$REMOTE_LOGIN72.COM file in the SYS$COMMON:[SYSEXE] directory and insert the appropriate commands.

Refer to Section 2.4.7 for information on how to access remote databases in a multiversion environment.

3.8.3.2 TCP/IP and the RDBSERVER Object

For TCP/IP, the Oracle Rdb Release 7.2 installation procedure defines RDB$REMOTE72 as the default account for the TCP/IP RDBSERVER object if the UCX utility is installed at that time. If UCX is not present when Oracle Rdb is installed, you must manually define the RDBSERVER object in UCX. See Section 4.1.2 for an explanation of setting up TCP/IP services for remote access.

3.8.4 Setting up Hot Standby

The following sections may be applicable if you have chosen to install the Hot Standby option. For more information on the Hot Standby feature, see the Oracle Rdb and Oracle CODASYL DBMS: Guide to Hot Standby Databases.

3.8.4.1 Network Accounts

Because the Hot Standby functionality requires a network object server (RDMAIJ72) to facilitate communications between the master and the standby database, the Hot Standby software automatically creates an RDMAIJ72 account and object. The installation procedure asks you to supply a valid user identifier for this account.

3.8.4.2 Network Protocols

You can specify either DECnet or TCP/IP as the network protocol, as described in the following sections.

3.8.4.2.1 DECnet

When you install your database software and choose the Hot Standby option, the installation procedure automatically configures the DECnet images necessary to use the Hot Standby capability. You do not need to perform any special tasks to install or invoke the DECnet network protocol.

3.8.4.2.2 TCP/IP

The TCP/IP network protocol is also supported, but is not automatically installed. To enable Hot Standby over a TCP/IP network, you must perform the following steps on both the master and standby nodes:

1. Define the RDMAIJ72 service:

   ```
   $TCP/IP SET SERVICE RDMAIJ72
       /PORT=n
       /USER_NAME=RDMAIJ72
       /PROCESS_NAME=RDMAIJ72
       /FILE=SYS$SYSTEM:rdmaijserver72.com
       /LIMIT=y
   ```

   where n is an available port number, and y is the number of connections permitted for the network service. A minimum of two connections is required for each database. In addition, any database recovery process (DBR) that executes on the master database also requires a connection.
2. Enable the service:

   ```
   $TCP/IP ENABLE SERVICE RDMAIJ72
   ```
3. Use the Transport qualifier with the RMU Replicate After Start or RMU Replicate Configure command to specify the network transport. The valid values for the Transport qualifier are DECNET and TCPIP.

   ```
   $RMU/REPLICATE AFTER CONFIGURE /TRANSPORT=TCPIP −
   _$ /STANDBY=NODE1:::DEV:[DIR]STANDBY_DB M_TESTDB
   ```

3.8.4.3 Privileges

For security reasons, the AIJSERVER account (RDMAIJ72) is created with just NETMBX and TMPMBX privileges. In most cases, these privileges are sufficient to start Hot Standby. However, for production Hot Standby systems, these privileges are not adequate to ensure continued replication in all environments and workload situations. Oracle recommends that you provide the following additional privileges for the AIJSERVER account:

- ALTPRI − This privilege allows the AIJSERVER to adjust its own priority to ensure adequate quorum (CPU utilization) for prompt message processing.
- PSWAPM − This privilege allows the AIJSERVER to enable and disable process swapping, which is also necessary to ensure prompt message processing.
- SETPRV − This privilege allows the AIJSERVER to temporarily set any additional privileges it may need to access the standby database or its server processes.
- SYSPRV − This privilege allows the AIJSERVER to access the standby database root file, if necessary.
- WORLD − This privilege allows the AIJSERVER to more accurately detect standby database server process failure and handle network failure more reliably.

3.8.5 Setting Up Cluster−Wide Statistics

You can use the RMU Show Statistics command with the Cluster qualifier to collect statistical data from an entire cluster. The Show Statistics command uses the account RDMSTT72 to collect statistical data from the nodes in the cluster. This account is created during installation of Oracle Rdb. It is created with the SYSPRV privilege, so it should have no problems accessing the database.

You can specify either DECnet or TCP/IP as the network protocol, as described in the following sections.

3.8.5.1 DECnet

The default transport mechanism used to communicate with the cluster members is DECnet; however, the TCP/IP network protocol is also supported.

3.8.5.2 TCP/IP

The TCP/IP network protocol is not automatically installed. To enable cluster statistics collection over a TCP/IP network, you must perform the following steps:

1. Define the RDMSTT72 service:

   ```
   $TCPIP SET SERVICE RDMSTT72
       /PORT=n
       /USER_NAME=RDMSTT72
       /PROCESS_NAME=RDMSTT72
       /FILE=SYS$SYSTEM:RDMSTTSERVER72.COM
       /LIMIT=y
   ```

   where n is an available port number, and y is the number of concurrent connections.
2. Enable the service:

   ```
   $TCP/IP ENABLE SERVICE RDMSTT72
   ```
3. Define RDMSST_NETWORK_TRANSPORT on the node where you will execute the RMU/SHOW STATISTICS/CLUSTER command:

   ```
   $DEFINE/SYSTEM RDM$BIND_STT_NETWORK_TRANSPORT "TCPIP"
   ```

   To switch back to the DECnet transport, deassign the RDM$BIND_STT_NETWORK_TRANSPORT logical name, or define it to be DECnet.

3.8.6 Displaying a List of Files Installed by Oracle Rdb

A file is written to your system that identifies all Oracle Rdb files installed on your system. To obtain this list after the installation ends, print or display a copy of the following file:

```
SYS$COMMON:[SYSMGR.VAXINFO$PRODUCTS]RDB072_72_FILES.DAT
```

3.9 Installing Oracle Rdb Images as Resident

You may improve the performance of applications using Oracle Rdb by installing several of the Oracle Rdb images as resident with the OpenVMS Install utility (INSTALL). Installing images as resident allows them to take advantage of several OpenVMS performance features.

The code sections of an image installed as resident reside in huge pages called granularity hint regions (GHRs) in memory. The OpenVMS Itanium and Alpha hardware environments can consider a set of pages as a single GHR. This GHR can be mapped by a single page table entry (PTE) in the translation buffer (TB). The result is a reduction in TB miss rates.

Furthermore, OpenVMS Alpha supports resource affinity domains (RADs) for certain hardware configurations. When RAD support is enabled, OpenVMS can replicate image data on each RAD. The advantage to this replication is that any CPU access to the image memory will always be in the same RAD.

To take advantage of this image data replication capability, the image must be installed in the system startup procedure before the end of SYSTARTUP_VMS.COM. The easiest way to accomplish this for the Oracle Rdb images is to execute SYS$STARTUP:RMONSTART72.COM from SYSTARTUP_VMS.COM (the site−specific system startup procedure).

If you use many resident images, you may need to modify the GH_RES_CODE system parameter to add at least 2048 additional pages. The System Dump Analyzer (SDA) command CLUE MEMORY/GH/FULL can be used to display the contents and free space within the Resident Image Code Regions.

To install several of the images as resident, pass the parameter "/RESIDENT" to the procedures RMONSTART72.COM and SQL$STARTUP.COM located in the SYS$STARTUP directory.

3.10 Oracle CDD/Repository Installed but Not Started Prior to Installation

If CDD/Repository is already installed on your system but not started, the IVP displays a message stating that the Oracle CDD/Repository is not started and that the test will be skipped. If you want to run the Oracle CDD/Repository test during the IVP, start Oracle CDD/Repository and rerun the IVP. Use the following command to start Oracle CDD/Repository:

```
$ @SYS$STARTUP:CDDSTRTUP
```

3.11 Running the IVP Separately

The Oracle Rdb Installation Verification Procedure (IVP) can be run at any time after the successful installation of Oracle Rdb. For example, if Oracle Rdb does not appear to be running properly, you may want to verify that the correct Oracle Rdb installation kit files are present on your system.

The account you use to run the IVP must have the TMPMBX and SYSPRV privileges. Also, the account quotas must be sufficient to run Oracle Rdb. Although you must execute the IVP from an account having the SYSPRV privilege, the installation kit files are provided with the protection of world−read and world−execute (W:RE). These protections allow nonprivileged users the ability to examine and copy these files.

To run the Oracle Rdb IVP after the installation of Oracle Rdb:

1. Set default to the SYS$COMMON:[SYSTEST] directory.
2. Invoke the IVP:

```
$ @RDB$IVP72
```

If the IVP fails, it creates a log file, SYS$UPDATE:RDBIVP.LOG, of the failed portion of the test.

3.12 Returning Read−Only Storage Areas to Original Settings

To return read−only storage areas to their original settings, enter the appropriate commands. For example:

```
SQL> ALTER DATABASE FILENAME MY_DB
cont> ALTER STORAGE AREA ARCHIVE READ ONLY;
```

3.13 Deleting Versions of Oracle Rdb

For your convenience, Oracle Rdb provides a command procedure, SYS$MANAGER:RDB$DEINSTALL_DELETE.COM, to delete current or previous versions of Oracle Rdb. You must run this command file from an account that has SETPRV privileges, or from an account that has SYSPRV, CMKRNL, SYSNAM, and WORLD privileges.

Note

As a precaution, back up your system disk before running the RDB$DEINSTALL_DELETE.COM command procedure.

You can use this command file if, for example, you decide to convert your production and repository databases to the latest version of Oracle Rdb and you want to delete a previous version or versions back to and including release 4.0.

Note

This procedure deletes SQL/Services as well as Oracle Rdb, even though SQL/Services is separately installed.

When you run the command file, you can optionally pass a single parameter that indicates the output location for all messages generated while the command file processes. This parameter can either be the name of a file (for example, RDB$DEINSTALL_DELETE.LOG) or the logical name SYS$OUTPUT (which displays messages on your screen).

To run the RDB$DEINSTALL_DELETE command procedure and have messages sent to a file named RDB$DEINSTALL_DELETE.LOG, enter the following command:

```
$ @SYS$MANAGER:RDB$DEINSTALL_DELETE.COM RDB$DEINSTALL_DELETE.LOG
```

Note

The RDBVMS$DEINSTALL_DELETE deinstallation command procedure provided in versions prior to V6.0 of Oracle Rdb is obsolete. Use the RDB$DEINSTALL_DELETE command procedure. In addition, note that the parameter passed with the RDBVMS$DEINSTALL_DELETE command procedure was the version to be deleted. This parameter is not valid for the new version of the deinstallation command procedure because the new version is menu−driven.

The command procedure checks for the existence of the different versions of Oracle Rdb on your system, and then displays a menu listing each version found (standard first, and then the oldest to the most current multiversion):

```
*************************************************************************
Rdb versions currently installed on your system

  1     Version 4.2 (Standard)
  2     Version 4.2 (Multiversion)
  3     Version 5.0 (Multiversion)
  4     Version 5.1 (Multiversion)
  5     Version 6.0 (Multiversion)
  6     Version 6.1 (Multiversion)
  0     Quit

Enter Choice to deinstall (0...6) :
```

If the command procedure displays an asterisk (*) next to a version entry on the menu, it means that version cannot be deleted by RDB$DEINSTALL_DELETE.COM because it is pre−release 4.0.

Enter the menu number for the version you want to delete. For example, to delete release 4.2 Multiversion, enter the following:

```
Enter Choice to deinstall (0...6) : 2
```

The command procedure displays the following message:

```
You are about to deinstall Rdb 4.2 (Multiversion)
```

If your system (for this example, named SYSTEM1) is a cluster member, the command procedure displays the following message and prompt:

```
This procedure will delete RMONSTOP42.COM.
If the Rdb Version 4.2 (Multiversion) monitor is running on any other node on
your cluster besides the node SYSTEM1, you will have to manually
stop the monitor on each of these other nodes after this
procedure has finished.
Do you want to check if the Rdb Version 4.2 (Multiversion) monitor is
currently running on your cluster? [N]:
```

If you enter YES, the command procedure checks each node in the cluster to see if the Oracle Rdb monitor or SQL/Services server for release 4.2 (Multiversion) is installed on that node, and displays an informational message similar to the following for each node found:

```
SQLSERVER started on node SYSTEM3

Rdb Version 4.2 (Multiversion) monitor started on node SYSTEM3
```

Regardless of whether you enter YES or NO, the command procedure creates the RDB$CLUSTER_DEINSTALL42.COM command procedure in your SYS$SCRATCH directory. Use this command procedure to deinstall Oracle Rdb Release 4.2 (Multiversion) from other nodes in the cluster. You must either run this command procedure on each node that has release 4.2 (Multiversion) installed, or use SYSMAN to run it clusterwide.

Next, the command procedure asks you to confirm that you want to continue with the deinstallation (whether or not your system is part of a cluster):

```
Enter Y(ES) to continue to deinstall Rdb 4.2 (Multiversion): YES
```

The final prompt asks you whether or not you want to delete the RDB$REMOTE42 account for the version you specified (keep this account if, for example, you plan to use it as a template to build other accounts):

```
Do you want to delete RDB$REMOTE42? [N]: YES
```

The command procedure takes five to ten minutes to complete the deletion of the appropriate files. It is complete when it displays the following message:

```
%RDB−I−END Deinstallation of Rdb 4.2 (Multiversion) now complete
```

3.14 Determining and Reporting Problems

If an error occurs while Oracle Rdb is being used and you believe that the error is caused by a problem with Oracle Rdb, contact your Oracle support representative. If you find an error in the Oracle Rdb documentation, please file a Bug so that it can be addressed.

Chapter 4
Using Remote Databases

Oracle Rdb allows access to databases that reside on remote nodes. A remote node refers to a computer system other than the one on which your application program or terminal session resides. Thus, remote access refers to the ability of a program on one node to communicate with a database system on a remote node.

For example, your company might want to use remote access because it has several warehouses located in different areas, each with its own inventory database. When a customer places an order and the local warehouse does not have the item in stock, you can access the inventory database of the other warehouses to find out if they have the item in stock. The remote access feature provides this kind of capability. This chapter describes how to:

- Set up the Oracle Rdb system to allow remote database access
- Grant database privileges for remote and network access
- Improve remote access performance
- Troubleshoot a remote database environment

For a description of accessing databases on remote systems after Oracle Rdb has been set up, see the Oracle Rdb7 Guide to SQL Programming.

4.1 Setting Up the System for Remote Access

Remote access makes it possible for a database on a remote node to be accessed as if it were local to the node. This can be useful even within a cluster to allow a database to be open on a single node in the cluster, for example, to optimize memory use for row caching. It also makes it possible to access a database with an earlier version of Oracle Rdb, including on the same node.

The Oracle Rdb installation automatically creates the RDB$REMOTE72 server account to allow remote access to Oracle Rdb databases. The RDB$REMOTE72 account can be used by any program accessing any remote database on OpenVMS.

TCP/IP support was added in release 6.1. Programs that execute on remote nodes can use TCP/IP to access Oracle Rdb Release 6.1 and higher databases. To access databases prior to release 6.1, you must use DECnet.

The Oracle Rdb Release 7.2 installation attempts to set up a service for TCP/IP only if UCX is found on your system. If you are using a TCP/IP product other than UCX, refer to the product documentation for information on setting up a service for Oracle Rdb. This section describes how to:

- Set up DECnet Phase IV, DECnet−Plus, and TCP/IP for remote access to Oracle Rdb on OpenVMS
- Verify the setup of the RDB$REMOTE72 account with the OpenVMS Authorize (AUTHORIZE) utility
- Enable the RDB$REMOTE72 account in the OpenVMS Authorize utility

4.1.1 Setting Up Remote Access in DECnet Phase IV

You must have the RDB$REMOTE72 account and object number 35 (RDBSERVER) in the Network Control Program (NCP) utility for proper functioning of Oracle Rdb remote features. This is needed on the node where the database resides and on the client. To ensure successful access to remote databases, verify that:

1. The RDBSERVER DECnet object exists. Use the NCP utility. See Section 4.1.1.1.
2. The password of the RDB$REMOTE account matches the password of the RDBSERVER DECnet object. See Section 4.1.1.2.
3. The RDB$REMOTE72 account exists. Use the OpenVMS Authorize utility (AUTHORIZE). See Section 4.1.3.

The verification steps listed here are explained in the following sections.

4.1.1.1 Verifying the RDBSERVER DECnet Object in the Network Control Program (NCP) Utility

To determine if the RDBSERVER DECnet object number 35 (RDBSERVER.COM) is present in the NCP utility, type the following commands:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN NCP

NCP> SHOW OBJECT RDBSERVER
```

```
Object Volatile Summary as of 23−MAY−2001 12:59:04

   Object   Number  File/PID                  User Id          Password

  RDBSERVER     35  RDBSERVER.COM             RDB$REMOTE       JUSTTESTING

NCP> EXIT
```

If the RDBSERVER DECnet object does not exist, you must install Oracle Rdb. Refer to Chapter 2 for installation procedures.

To allow a remote node access to a database on your system, set the proxy access for the RDBSERVER DECnet object to incoming using the NCP utility. To access a database on a remote node, set the proxy access to outgoing. Allowing access to and from your system is the default.

To verify the status of proxy access, type the following commands:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN NCP
NCP> SHOW OBJECT RDBSERVER CHARACTERISTICS

Object Volatile Characteristics as of 23−MAY−2001 13:01:05

Object = RDBSERVER

Number                   = 35
File id                  = RDBSERVER.COM
User id                  = RDB$REMOTE72
Account                  = RDB$REMOTE72
Password                 = JUSTTESTING
Proxy access             = incoming and outgoing
```

To change the status of the proxy access to only incoming, type the following command:

```
NCP> SET OBJECT RDBSERVER PROXY INCOMING
```

To change the status of the proxy access to only outgoing, type the following command:

```
NCP> SET OBJECT RDBSERVER PROXY OUTGOING
```

To set the status of proxy access to both incoming and outgoing, type the following command:

```
NCP> SET OBJECT RDBSERVER PROXY BOTH
```

If you are working on a cluster system or if someone is accessing your cluster system from a remote node, be sure the proxy access is set correctly on each node. Do not use the cluster alias name.

Check the OpenVMS file protections on the SYS$SYSTEM:RDBSERVER72.EXE and SYS$SYSTEM:RDBSERVER.COM files. They should both be assigned WORLD READ and EXECUTE privileges. If these privileges are not set, RDBSERVER cannot run and remote access fails.

4.1.1.2 Verifying Matching Passwords for the RDB$REMOTE72 Account in UAF and for the RDBSERVER DECnet Object in the NCP Utility

The password for the RDB$REMOTE72 account in the user authorization file (UAF) must be the same as the password for the RDBSERVER DECnet object in the Network Control Program (NCP) utility.

If the passwords are different, then any remote operation will fail. Therefore, you must update the passwords in two places: the UAF and NCP.

Simply looking at the password for the RDBSERVER DECnet object in the NCP utility and then setting the RDB$REMOTE72 password in UAF to the same thing does not guarantee a match. You must reset the password in both places to ensure a match. Type the following commands:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE

UAF> MODIFY RDB$REMOTE72/PASSWORD=password
UAF> EXIT
%UAF−I−DONEMSG, system authorization file modified
%UAF−I−NAFNOMODS, no modifications made to network proxy data base
%UAF−I−RDBNOMODS, no modifications made to rights data base

$ SET DEFAULT SYS$SYSTEM
$ RUN NCP

NCP> SET OBJECT RDBSERVER PASSWORD password
NCP> DEFINE OBJECT RDBSERVER PASSWORD password
NCP> EXIT
```

To permanently change the password in the NCP utility, you must do the two−step procedure shown in the preceding example. The SET statement changes the password in the volatile database, and the DEFINE statement changes it in the permanent database.

If you are working on a cluster system or if someone is accessing your cluster system from a remote node, be sure that each node has the same password for the RDB$REMOTE72 account and RDBSERVER DECnet object.

4.1.1.3 Setting Up Remote Access in DECnet−Plus

You must have the RDB$REMOTE72 account and object number 35 (RDBSERVER) in the Network Control Language (NCL) utility for proper functioning of Oracle Rdb remote server features. To ensure successful access to remote databases, verify that:

1. The RDB$REMOTE72 account exists. Use the OpenVMS Authorize (AUTHORIZE) utility. Section 4.1.3 provides more detail about the RDB$REMOTE72 account.
2. The RDB$REMOTE72 account is enabled.
3. The RDBSERVER DECnet object number 35 is present in the NCL utility.
   If the RDBSERVER DECnet object does not exist, you must install Oracle Rdb. Section 4.1.1.4 explains how to verify if the DECnet object is present. Refer to Chapter 2 for installation procedures.

4. The status of proxy access is appropriate.
   To allow remote node access to a database on your system, set the proxy access for the RDBSERVER DECnet object to incoming using the NCL utility. To allow access to a database on a remote node, set the proxy access to outgoing. Allowing access to and from your system is the default. Section 4.1.1.4 and Section 4.1.1.5 explain how to check and change the status of the proxy access.
5. Database privileges exist for RDB$REMOTE72. Section 4.2.1 describes how to grant database privileges for remote access.
6. That proxy accounts are set up to avoid displaying the RDB$REMOTE72 password. The Oracle Rdb7 Guide to SQL Programming describes how to attach to a remote database through a proxy account.
7. That the LOGIN.COM procedures for the RDB$REMOTE72 account and any proxy accounts contain the appropriate commands if you want product−specific files to be run during the RDB$REMOTE72 login step. Section 3.8.3 and Section 2.4.7 provide information on RDB$REMOTE_LOGIN72.COM and LOGIN.COM procedures for proxy accounts.

4.1.1.4 Verifying the Status of the DECnet Object and Proxy Access

To verify both the presence of the RDBSERVER DECnet object and the status of proxy access, you can use a single NCL utility SHOW NODE command. The following NCL utility example shows that the RDBSERVER DECnet object number 35 is present in the NCL database and that proxy access is set to both incoming and outgoing:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN NCL
NCL>SHOW NODE 0 SESSION CONTROL APPLICATION RDBSERVER ALL CHARACTERISTICS

Node 0 Session Control Application RDBSERVER
at 2001−01−09−16:33:28.790−04:00Iinf

Characteristics

    Client                            = <Default value>
    Addresses                         =
       {
          name = RDBSERVER ,
          number = 35
       }
    Outgoing Proxy                    = True
    Incoming Proxy                    = True
    Outgoing Alias                    = True
    Incoming Alias                    = True
    Node Synonym                      = True
    Image Name                        = SYS$SYSTEM:RDBSERVER.COM
    User Name                         = "RDB$REMOTE72"
    Incoming OSI TSEL                 = <Default value>
NCL> EXIT
```

4.1.1.5 Changing the Status of the Proxy Access

If you want to change the status of the proxy access to incoming only, type the following command:

```
NCL>SET NODE 0 SESSION CONTROL APPLICATION RDBSERVER INCOMING PROXY = TRUE
```

```
Node 0 Session Control Application RDBSERVER
at 2001−01−09−08:50:16.490−04:00Iinf

Characteristics

    Incoming Proxy                    = True
```

If you want to change the status of the proxy access to outgoing only, type the following command:

```
NCL>SET NODE 0 SESSION CONTROL APPLICATION RDBSERVER OUTGOING PROXY = TRUE
Node 0 Session Control Application RDBSERVER
at 2001−01−09−08:50:36.320−04:00Iinf

Characteristics

    Outgoing Proxy                    = True
```

Refer to the DECnet−Plus documentation for more information about making these types of settings.

If you are working on a cluster system or if someone is accessing your cluster system from a remote node, be sure the proxy access is set correctly on each node. Do not use the cluster alias name.

Check the OpenVMS file protections on the SYS$SYSTEM:RDBSERVER.EXE and SYS$SYSTEM:RDBSERVER.COM files. They should both be assigned WORLD READ and EXECUTE privileges. If these privileges are not set, RDBSERVER cannot run and remote access fails.

4.1.2 Setting Up Remote Access in TCP/IP Services

The TCP/IP network protocols can be used to access remote Release 6.1 and higher databases. To do this, you must have the TCP/IP service RDBSERVER defined with the UCX utility. The Oracle Rdb installation procedure will automatically set up and enable the RDBSERVER service if the UCX utility is installed and started. If the installation cannot set up the service, you will need to set up the RDBSERVER service manually.

To ensure successful access to databases from remote systems, verify:

1. The existence of the RDB$REMOTE72 account using the OpenVMS Authorize (AUTHORIZE) utility. Section 4.1.3 provides more detail about the RDB$REMOTE72 account.
2. The presence of the RDBSERVER service in the UCX utility.

4.1.2.1 Verify the Presence of the RDBSERVER Service

To verify the presence of the RDBSERVER service, you use the UCX utility SHOW SERVICE command. The RDBSERVER service must be enabled if the SHOW SERVICE command is to display full statistics. The following example shows that the service is present, enabled, and is using port 611, account RDB$REMOTE72, and file SYS$SYSTEM:RDBSERVER.COM.

```
$ SET DEFAULT SYS$SYSTEM
$ UCX
UCX> show service rdbserver/full
```

```
Service: RDBSERVER
                           State:     Enabled
Port:              611     Protocol:  TCP             Address:  0.0.0.0
Inactivity:          5     User_name: RDB$REMOTE72    Process:  RDB
Limit:              10     Active:      0             Peak:       0

File:         SYS$SYSTEM:RDBSERVER.COM
Flags:        Listen

Socket Opts:  Rcheck Scheck
 Receive:            0     Send:               0

Log Opts:     None
 File:        not defined

Security
 Reject msg:  not defined
 Accept host: 0.0.0.0
 Accept netw: 0.0.0.0
```

4.1.2.2 Set Up the RDBSERVER Service

If the RDBSERVER service does not exist, set up the service as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ UCX
UCX> SET SERVICE
RDBSERVER/PORT=611/USER_NAME=RDB$REMOTE72/PROCESS=RDB72/LIMIT=10/−
_UCX>FILE=SYS$SYSTEM:RDBSERVER.COM
UCX> ENABLE SERVICE RDBSERVER
```

The value for /LIMIT must be greater than the expected number of simultaneous connects. For more information, see Section 4.1.2.3.

To use TCP/IP for remote access on another node that shares the cluster common root directory, you must enable the UCX utility service RDBSERVER on that node. Log in to each node and do the following:

```
UCX> enable service rdbserver
```

Refer to the TCP/IP Services for OpenVMS documentation for more information about the UCX utility.

Check the OpenVMS file protections on the SYS$SYSTEM:RDBSERVERnn.EXE (where nn could be an Oracle Rdb release number) and SYS$SYSTEM:RDBSERVER.COM files. They should both be assigned WORLD READ and EXECUTE privileges. If these privileges are not set, RDBSERVER cannot run and remote access fails.

4.1.2.3 Changing the UCX /LIMIT Defaults

On a given OpenVMS node running TCP/IP, the UCX /LIMIT value for the RDBSERVER service determines the number of simultaneous remote attachments over one link that are possible to Oracle Rdb databases on that node. Each remote attachment through TCP/IP may create its own process. The default value established by the Oracle Rdb installation for the /LIMIT value is 10.

It may be necessary to customize this value for your installation. Decrease this number if the possibility of ten RDBSERVER processes is excessive for your system. Increase this value if you expect workloads requiring more than ten simultaneous attaches to Oracle Rdb databases on your system. If this value is increased substantially, you should adjust the MAXPROCESSCNT SYSGEN parameter to account for the possible creation of multiple RDBSERVER processes.

To change the /LIMIT value for the RDBSERVER service in UCX, log into a privileged account and issue the following commands:

```
$ UCX
UCX> DISABLE SERVICE RDBSERVER
UCX> SET SERVICE RDBSERVER /LIMIT=20
UCX> ENABLE SERVICE RDBSERVER
UCX> EXIT
```

4.1.3 Verifying the Setup of the RDB$REMOTE Account with the OpenVMS Authorize Utility

Use the OpenVMS Authorize (AUTHORIZE) utility to determine if the RDB$REMOTE72 account exists on your system. You must have the system user identification code (UIC) or the SYSPRV privilege to run AUTHORIZE. For example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW RDB$REMOTE72

Username: RDB$REMOTE72                       Owner:
Account:                                   UIC:    [10,1] ([RDB$REMOTE72])
CLI:      DCL                              Tables: DCLTABLES
Default:  SYS$COMMON:[RDB$REMOTE72]

LGICMD:   SYS$SYSTEM:RDB$REMOTE_LOGIN72.COM
Flags:  Disctly DefCLI Lockpwd Dismail Disreconnect
   .
   .
   .
```

If the RDB$REMOTE72 account does not exist, you must install Oracle Rdb. Refer to Chapter 2 for installation procedures.

4.2 Granting Database Privileges for Remote and Network Access

This section describes how to grant database privileges to the RDB$REMOTE72 account for remote access and to the NETWORK identifier for network access.

Under the Oracle Rdb default protection scheme, when you create a new database, table, or view, you (as its owner) get all access rights (privileges) to that database or object. Getting access rights means that Oracle Rdb creates an entry to the Oracle Rdb access privilege set, called the access control list (ACL), for the database, table, or view. Each entry in an ACL consists of an identifier and a list of privileges assigned to the identifier:

- Each identifier specifies a user or a set of users.
- The list of privileges specifies what operations that user or user group can perform on the database, table, or view.

In this example, Oracle Rdb associates an identifier ([SQL,AARON]) with a list of privileges (ACCESS=SELECT ...):

```
SQL> ATTACH 'FILENAME PERSONNEL';
SQL> SHOW ALIAS;
Default alias:
    Oracle Rdb database in file PERSONNEL.RDB
SQL> SHOW PROTECTION ON DATABASE RDB$DBHANDLE;
Protection on Alias RDB$DBHANDLE
    (IDENTIFIER=[SQL,AARON],ACCESS=SELECT+INSERT+UPDATE+DELETE+SHOW+CREATE+
      ALTER+DROP+DBCTRL+OPERATOR+DBADM+SECURITY+DISTRIBTRAN)
    (IDENTIFIER=[*,*],ACCESS=NONE)
```

In effect, Oracle Rdb associates your user identification code (UIC), called an identifier, with a list of database privileges when you create a database, table, or view.

However, when Oracle Rdb creates a database it does not automatically give the RDB$REMOTE72 account any access rights to it. Thus, to enable a database for remote access, you must grant it database privileges with the GRANT statement. See Section 4.2.1 for information about granting database privileges to the RDB$REMOTE72 account to allow remote access to a database.

While Oracle Rdb does not implicitly grant any database privileges to the RDB$REMOTE72 account, it does implicitly grant all database privileges to the NETWORK identifier. Like the RDB$REMOTE72 account, the NETWORK identifier must have an entry in a database's ACL for that database to be accessed remotely. Both identifiers must exist in a database's ACL for remote access to occur. See Section 4.2.2 for information about controlling privileges for the NETWORK identifier for network access.

See the Oracle Rdb7 SQL Reference Manual for more information on the SQL GRANT and SQL REVOKE statements. See the Oracle Rdb7 Guide to Database Design and Definition for more information on access control lists (ACLs).

4.2.1 Granting Database Privileges to the RDB$REMOTE72 Account for Remote Access

Oracle Rdb does not give the RDB$REMOTE72 account any database privileges when a database is created. To enable a database for remote access, you must grant it privileges explicitly. For example, to grant the RDB$REMOTE72 account the SELECT privilege only, type:

```
$ SQL
SQL> ATTACH 'FILENAME NODEB::PERSONNEL.RDB';
SQL> SHOW PROTECTION ON DATABASE RDB$DBHANDLE;

Protection on Alias RDB$DBHANDLE
    (IDENTIFIER=[SQL,SMITH],ACCESS=SELECT+INSERT+UPDATE+DELETE+SHOW+CREATE+
      ALTER+DROP+DBCTRL+OPERATOR+DBADM+REFERENCES+SECURITY+DISTRIBTRAN)
    (IDENTIFIER=[*,*],ACCESS=NONE)

SQL> GRANT SELECT ON DATABASE RDB$DBHANDLE TO RDB$REMOTE72;
SQL> SHOW PROTECTION ON DATABASE RDB$DBHANDLE;

Protection on Alias RDB$DBHANDLE
    (IDENTIFIER=[RDB$REMOTE72],ACCESS=SELECT)
    (IDENTIFIER=[SQL,SMITH],ACCESS=SELECT+INSERT+UPDATE+DELETE+SHOW+CREATE+
      ALTER+DROP+DBCTRL+OPERATOR+DBADM+REFERENCES+SECURITY+DISTRIBTRAN)
    (IDENTIFIER=[*,*],ACCESS=SELECT)
SQL> COMMIT;
SQL> DISCONNECT DEFAULT;
SQL> EXIT
```

By default, the RDB$REMOTE72 account is not a privileged account. When you grant database privileges to the remote account for the PERSONNEL database, you are, in effect, allowing anyone remote access to that database.

4.2.2 Controlling Database Privileges for Network Access

Oracle Rdb automatically grants all database privileges to the NETWORK identifier for every database it creates. Thus, you do not have to grant privileges for a database to make it accessible remotely. However, you might want to reduce the number of privileges that you allow for a database. For example, to grant the SELECT privilege only, type:

```
SQL> GRANT SELECT ON DATABASE A TO NETWORK;
SQL> SHOW PROTECTION ON DATABASE A;
Protection on Alias A
    (IDENTIFIER=NETWORK,ACCESS=SELECT)
    (IDENTIFIER=[RDB,RDB_EXECUTE],ACCESS=SELECT+INSERT+UPDATE+DELETE+SHOW+
      CREATE+ALTER+DROP+DBCTRL+OPERATOR+DBADM+REFERENCES)
    (IDENTIFIER=[*,*],ACCESS=SELECT+INSERT+UPDATE+DELETE+SHOW+CREATE+ALTER+
      DROP+OPERATOR+DBADM+REFERENCES)
SQL> COMMIT;
SQL> DISCONNECT DEFAULT;
SQL> EXIT
```

After you commit the statement, remote users will only be able to select data from Database A; they will not be able to perform any other operations.

Note

Because Oracle Rdb implicitly grants the NETWORK identifier all database privileges, issuing a SHOW PROTECTION statement on a database does not reveal an entry for it. Only after you have explicitly granted (or granted and then revoked) privileges to the NETWORK identifier can you see it as an ACL entry.

4.2.3 Enabling File System Access to Database Files

When you attach to a remote Oracle Rdb database, the remote operating system sees you as the server account. The directory containing the Oracle Rdb database and all of its parent directories must, at least, permit read access to the server account. Without read access, attempts to attach to remote databases in that directory fail with file protection errors. Note the distinction between the server account and the account specified in a USER ... USING clause of an SQL CREATE DATABASE or SQL ATTACH statement. You are seen as the server account on the remote node as far as the remote operating system is concerned.

This remains true for the duration of your remote session. However, internally to Oracle Rdb, if you specified a USER ... USING clause, you take on that identity within Oracle Rdb only for the purpose of internal database security checks.

4.3 Improving Performance When Attaching to a Remote Database

The following sections discuss how you can increase network performance when connecting to a remote database.

4.3.1 Specifying Configuration Files to Improve Remote Access

Oracle Rdb provides two types of configuration files that you can use to improve network access to remote databases:

- Client configuration file
  You create a client configuration file for use on your client systems. You must name it RDB$CLIENT_DEFAULTS.DAT.
- Server configuration file
  You create a server configuration file for use on your server systems. You must name it RDB$SERVER_DEFAULTS.DAT.

Note

You can specify these configuration files for remote access only when Oracle Rdb Release 6.1 or later is installed on both client and server systems.

Table 4−1 shows the set of parameters that you can use in a client and server configuration file to configure network access to remote databases.

Table 4−1 Valid Parameters in Client and Server Configuration Files

Configuration File Type	Configuration File Name	Valid parameters
Client	RDB$CLIENT_DEFAULTS.DAT	♦ SQL_ALTERNATE_SERVICE_NAME ♦ SQL_DEFAULTS_RESTRICTION ♦ SQL_ENABLE_PROBE ♦ SQL_MESSAGE_VECTOR_RETURN_TYPE ♦ SQL_NETWORK_BUFFER_SIZE ♦ SQL_NETWORK_NUMBER_ATTACHES ♦ SQL_NETWORK_TRANSPORT_TYPE ♦ SQL_PASSWORD ♦ SQL_RCV_PREFETCH_ROWS ♦ SQL_SGS_PREFETCH_ROWS ♦ SQL_TRANS_START_WAIT ♦ SQL_USERNAME
Server	RDB$SERVER_DEFAULTS.DAT	♦ SQL_ALTERNATE_SERVICE_NAME ♦ SQL_DEFAULTS_RESTRICTION ♦ SQL_NETWORK_BUFFER_SIZE

The SQL_ALTERNATE_SERVICE_NAME, SQL_DEFAULTS_RESTRICTION, and SQL_NETWORK_BUFFER_SIZE parameters are called common parameters because both a client and a server configuration file can include them. In contrast, the other parameters listed for the client are valid in a client configuration file only. At installation time, Oracle Rdb internally sets a default value for each of the parameters listed in Table 4−2.

Table 4−2 Summary of Configuration File Parameters and Their Defaults

Parameter	Acceptable Values	Default Value	Configuration File
SQL_ALTERNATE_SERVICE_NAME	text	RDBSERVER	Client or server
SQL_DEFAULTS_RESTRICTION	SYSTEM GROUP USER	USER	USER Client or server
SQL_ENABLE_PROBE	TRUE FALSE	FALSE	Client only
SQL_MESSAGE_VECTOR_RETURN_TYPE	TEXT STATUS INTERNAL	INTERNAL	Client only
SQL_NETWORK_BUFFER_SIZE	A numeric value in the range 500 to 64,000 bytes	4,096	Client or server
SQL_NETWORK_NUMBER_ATTACHES	A numeric value greater than zero	10	Client only
SQL_NETWORK_TRANSPORT_TYPE	TCPIP DECNET DEFAULT	DECNET	Client only
SQL_PASSWORD	text	none	Client only
SQL_RCV_PREFETCH_ROWS	A numeric value greater than zero	20	Client only
SQL_SGS_PREFETCH_ROWS	A numeric value greater than zero	20	Client only
SQL_TRANS_START_WAIT	numeric	3 seconds	Client only
SQL_USERNAME	text	none	Client only

Because Oracle Rdb has preset internal defaults for all configuration file parameters (except SQL_USERNAME and SQL_PASSWORD), you do not have to create any configuration files. However, configuration files provide flexibility that you might find useful as you try to control remote access for a wide variety of applications and user needs. Setting up configuration files enables a database administrator (DBA), system manager, or programmer to alter the preset, internal parameter default settings at the system logical, group logical, or (user) process logical level.

Oracle Rdb lets you create configuration files (as described in Section 4.3.2) in any of three separate directories pointed to by the following logical names:

- RDB$SYSTEM_DEFAULTS
  This logical name is defined in the system logical name table.
- RDB$GROUP_DEFAULTS
  This logical name is defined in the group logical name table.
- RDB$USER_DEFAULTS
  This logical name is defined in the process logical name table.

On the initial attach to a remote database, Oracle Rdb first checks the directory pointed to by the RDB$SYSTEM_DEFAULTS logical name. If it finds a configuration file, it reads the file to check the values assigned to the parameters that are specified. It first checks the SQL_DEFAULTS_RESTRICTION parameter, because that parameter determines whether Oracle Rdb also reads any other configuration files located in the directories defined by the RDB$GROUP_DEFAULTS and RDB$USER_DEFAULTS logical names. This occurs for both the client and the server.

If none of these logical names are defined, Oracle Rdb uses the SYS$LOGIN directory.

Suppose a database administrator created the following configuration file called RDB$CLIENT_DEFAULTS.DAT and put it in the RDB$SYSTEM_DEFAULTS directory:

```
SQL_DEFAULTS_RESTRICTION SYSTEM
SQL_NETWORK_BUFFER_SIZE  10100
SQL_RCV_PREFETCH_ROWS  50
```

The SYSTEM value signifies that you want Oracle Rdb to adjust the internal defaults using only the configuration file located in the RDB$SYSTEM_DEFAULTS directory, namely the configuration file that it has already read. After Oracle Rdb reads the system configuration file, it resets the internal defaults as illustrated in Table 4−3.

Table 4−3 Resetting Internal Parameter Defaults After Reading a System Configuration File

Parameter Name	Initial Preset Internal Default	Resulting Internal Default
SQL_ALTERNATE_SERVICE_NAME	RDBSERVER	RDBSERVER
SQL_DEFAULTS_RESTRICTION	USER	SYSTEM
SQL_ENABLE_PROBE	FALSE	FALSE
SQL_MESSAGE_VECTOR_RETURN_TYPE	INTERNAL	INTERNAL
SQL_NETWORK_BUFFER_SIZE	4096	10100
SQL_NETWORK_NUMBER_ATTACHES	10	10
SQL_NETWORK_TRANSPORT_TYPE	DECnet	DECnet
SQL_PASSWORD	none	none
SQL_RCV_PREFETCH_ROWS	20	50
SQL_SGS_PREFETCH_ROWS	20	20
SQL_USERNAME	none	none

As the table shows, Oracle Rdb changes the SQL_DEFAULTS_RESTRICTION parameter value from USER to SYSTEM, the SQL_NETWORK_BUFFER_SIZE parameter value from 4,096 to 10,100 bytes, and the SQL_RCV_PREFETCH_ROWS parameter value from 20 to 50. All other parameter values remain as they were initially set.

If the RDB$CLIENT_DEFAULTS.DAT configuration file that was put in the RDB$SYSTEM_DEFAULTS directory specified the GROUP value instead of SYSTEM as in the previous example, Oracle Rdb would have read the configuration file in the system logical directory and then read the configuration file located in the group logical directory. Whichever settings the group configuration file specifies override any equivalent settings specified in either the system configuration file or by the initial default settings. In general, the parameters explicitly set in the last read configuration file override all previously set parameters.

Thus, if the RDB$CLIENT_DEFAULTS.DAT configuration file specified USER instead of SYSTEM or GROUP as in the previous examples, Oracle Rdb would read the configuration file in the system logical directory, then the group logical directory, and finally the user logical directory. Any settings specified in the user configuration file would override any settings previously read.

You do not have to include a system configuration file. For example, you can include a group configuration file only to control parameter settings at the group logical level. You might want to include a group and a user configuration file or just a user configuration file to impose a mixture of group settings with process settings. Review the needs of your site to determine the configuration files that you want to create in the three configuration file directory locations.

The following sections describe how to create a configuration file and present reference information about the parameters that a configuration file can include.

4.3.2 Creating a Configuration File

To create an RDB$CLIENT_DEFAULTS.DAT or RDB$SERVER_DEFAULTS.DAT configuration file, invoke a text editor and type the parameter keyword, one or more spaces or TAB characters, and a single parameter value (on the same line). Parameter values for the following keywords must be specified in UPPER CASE: SQL_NETWORK_TRANSPORT_TYPE, SQL_DEFAULTS_RESTRICTION, and SQL_MESSAGE_VECTOR_RETURN_TYPE. For example, the following RDB$CLIENT_DEFAULTS.DAT client configuration file changes the defaults for three parameters:

```
SQL_DEFAULTS_RESTRICTION        SYSTEM
SQL_NETWORK_BUFFER_SIZE         10100
SQL_NETWORK_NUMBER_ATTACHES     5
```

The order of the parameters is not significant, but you might want to impose your own ordering rules to make reading configuration files easier. Oracle Rdb uses internal system default values when:

- ♦ You misspell a parameter name
- ♦ You specify an invalid parameter value

Note

Oracle Rdb does not warn you with an error message when you specify an invalid parameter value. Check your configuration file parameter values

carefully to ensure that remote access works as you expect.

- ♦ You omit a parameter
- ♦ You do not specify parameter values in UPPER CASE for the following keywords:
    - ◊ SQL_NETWORK_TRANSPORT_TYPE
    - ◊ SQL_DEFAULTS_RESTRICTION
    - ◊ SQL_MESSAGE_VECTOR_RETURN_TYPE

After you create a configuration file, put it in one of the three directory locations pointed to by the following Oracle Rdb assigned logical names:

- ♦ RDB$SYSTEM_DEFAULTS
- ♦ RDB$GROUP_DEFAULTS
- ♦ RDB$USER_DEFAULTS

4.3.2.1 Specifying SQL_ALTERNATE_SERVICE_NAME

When using the TCP/IP transport, you can use the SQL_ALTERNATE_SERVICE_NAME parameter to specify the name of an alternate UCX service for remote database access. This is especially useful if you need to access an earlier version of a database through TCP/IP (see Section 2.4.7.2 for details). This parameter can also be used for any other special access requirements that are not met by the default RDBSERVER UCX service.

Table 4−2 provides key information about the SQL_ALTERNATE_SERVICE_NAME parameter.

4.3.2.2 Specifying SQL_DEFAULTS_RESTRICTION

The SQL_DEFAULTS_RESTRICTION parameter controls the startup of network default characteristics for the system, group, or user. You can use the SQL_DEFAULTS_RESTRICTION parameter in a client or server configuration file (parameter must be in UPPER CASE).

Table 4−2 provides key information about the SQL_DEFAULTS_RESTRICTION parameter.

Oracle Rdb uses a set of defaults for the SYSTEM, GROUP, and USER values for the SQL_DEFAULTS_RESTRICTION parameter to control what configuration files Oracle Rdb reads when establishing parameter defaults during an Oracle Rdb remote operation.

Refer to Section 4.3.1 for detailed information about how Oracle Rdb uses the SQL_DEFAULTS_RESTRICTION parameter.

4.3.2.3 Specifying SQL_ENABLE_PROBE

The SQL_ENABLE_PROBE parameter turns on address verification so that all addresses passed to Oracle Rdb will be checked first to make sure they are pointing to memory locations with the appropriate protection. Valid values for SQL_ENABLE_PROBE are TRUE or FALSE.

Address probing is useful if a program gets segment violations and the program counter (PC) is pointing to Oracle Rdb. It may be that bad addresses are being passed to Oracle Rdb. Turning on the probe function can help pinpoint the bug in the calling program. Normally, probing is turned off, as there is a slight performance penalty for having it turned on.

Table 4−2 provides key information about the SQL_ENABLE_PROBE parameter.

4.3.2.4 Specifying SQL_MESSAGE_VECTOR_RETURN_TYPE

When a status is returned from the remote server, you occasionally receive a NONAME secondary error because the local system does not recognize the status code returned by the remote server. For example, a secondary error could be that the Oracle Rdb engine is not installed on the client system or that the remote system is not OpenVMS. To overcome this condition, you can set the SQL_MESSAGE_VECTOR_RETURN_TYPE parameter to TEXT (parameter must be in UPPER CASE).

The TEXT value translates all secondary error messages to text format on the remote server before the errors are returned to the client.

The STATUS value returns the secondary status error code. All statuses are returned exactly as they appear on the host system. They are not translated into text.

The default value of INTERNAL means that Oracle Rdb chooses the best return method for your configuration.

Table 4−2 provides key information about the SQL_MESSAGE_VECTOR_RETURN_TYPE parameter.

4.3.2.5 Specifying SQL_NETWORK_BUFFER_SIZE

The SQL_NETWORK_BUFFER_SIZE parameter defines the number of bytes to pack into one network buffer. If you transfer large amounts of data in or out of the database, you may want to increase the buffer size to improve performance. Increasing the buffer size reduces the number of network I/O operations used when large data transfers are made.

Suppose the size of a fetched row is 10,000 bytes. A buffer size of 5,000 bytes requires two network messages to transfer the 10,000−byte data row. A buffer size of 10,000 bytes takes only one network message. When calculating the network buffer size, however, be sure to add an extra 100 bytes to allow for the message header. For example, if you need a 10,000−byte network buffer size, specify 10,100 bytes.

You can use the SQL_NETWORK_BUFFER_SIZE parameter in a client or server configuration file. If you define SQL_NETWORK_BUFFER_SIZE in both the RDB$SERVER_DEFAULTS.DAT and RDB$CLIENT_DEFAULTS.DAT files, Oracle Rdb compares the values and picks the lower of the two.

Table 4−2 provides key information about the SQL_NETWORK_BUFFER_SIZE parameter.

If you change your network buffer size, be sure that your system and process quotas are sufficient to accommodate the change.

Note

> ***For compatibility with prior releases of Oracle Rdb, the RDB$REMOTE_BUFFER_SIZE logical name can still be defined in the current***

release for the network buffer size on client systems; however, if you define the SQL_NETWORK_BUFFER_SIZE parameter in a configuration file, its value overrides the value set for the RDB$REMOTE_BUFFER_SIZE logical name.

4.3.2.6 Specifying SQL_NETWORK_NUMBER_ATTACHES

The SQL_NETWORK_NUMBER_ATTACHES parameter signifies the maximum number of attaches that can be done across one logical network link.

Suppose there are 11 attaches, the SQL_NETWORK_NUMBER_ATTACHES parameter is set to 10, and the attaches are made to the same remote node. The 11th attach is made over a new logical link.

Table 4−2 provides key information about the SQL_NETWORK_NUMBER_ATTACHES parameter.

Note

For compatibility with prior releases of Oracle Rdb, the RDB$REMOTE_MULTIPLEX_OFF logical name is still valid in the current release; however, by enabling the RDB$REMOTE_MULTIPLEX_OFF logical name, you limit the number of network attaches to one. If you define the SQL_NETWORK_NUMBER_ATTACHES parameter, its value overrides the value set for the RDB$REMOTE_MULTIPLEX_OFF logical name.

4.3.2.7 Specifying SQL_NETWORK_TRANSPORT_TYPE

The SQL_NETWORK_TRANSPORT_TYPE parameter specifies the network protocol to be used to access a database on a remote system. Valid values for the SQL_NETWORK_TRANSPORT_TYPE parameter are TCPIP, DECNET, and DEFAULT.

To access an Oracle Rdb database on another system, your system and the system on which the database resides must both use the same communication protocol (both systems must use DECnet or both systems must use TCP/IP).

If your system has only one communication protocol (DECnet or TCP/IP) installed, you can attach to a database on another system that uses the same protocol. If you try to access a database on another system that uses a different protocol, the attempt fails.

A system can have more than one protocol installed. From a system that has both DECnet and TCP/IP installed, you can access a database on a remote system that uses either the DECnet or TCP/IP protocol. DECnet is the default communication protocol for an OpenVMS system that has both DECnet and TCP/IP installed. When you attempt to access a database on a remote system from an OpenVMS system, Oracle Rdb will first use DECnet. If the attempt fails using DECnet, Oracle Rdb automatically tries again using TCP/IP. If your OpenVMS system has both DECnet and TCP/IP installed and you want to use only one protocol for remote access, add a line to your RDB$CLIENT_DEFAULTS.DAT client configuration file that identifies the protocol to be used exclusively (the parameter must be in UPPER CASE):

```
! To use TCP/IP exclusively:
SQL_NETWORK_TRANSPORT_TYPE          TCPIP

       or

! To use DECnet exclusively:
SQL_NETWORK_TRANSPORT_TYPE          DECNET
```

If you have explicitly set the TCPIP or DECNET protocol in the RDB$CLIENT_DEFAULTS.DAT client configuration file at the system or group level, you can reset to the default behavior by changing the SQL_NETWORK_TRANSPORT_TYPE parameter to DEFAULT, as shown in the following example:

```
! To reset to the default behavior:
SQL_NETWORK_TRANSPORT_TYPE          DEFAULT
```

Table 4−2 provides key information about the SQL_NETWORK_TRANSPORT_TYPE parameter.

4.3.2.8 Specifying SQL_RCV_PREFETCH_ROWS

The SQL_RCV_PREFETCH_ROWS parameter controls the number of rows the database fetches all at once. These rows are sent to the client in as many network messages as are required.

Suppose you enter a SELECT wildcard statement (SELECT * ...) that returns 40 rows. The SQL_RCV_PREFETCH_ROWS parameter is set to 20. Two network messages are needed to complete the receive operation.

Table 4−2 provides key information about the SQL_RCV_PREFETCH_ROWS parameter.

4.3.2.9 Specifying SQL_SGS_PREFETCH_ROWS

The SQL_SGS_PREFETCH_ROWS parameter controls the number of prefetch get−segmented−string rows for each get−segmented−string message.

Suppose you want to fetch 40 segmented string rows but the SQL_SGS_PREFETCH_ROWS parameter is set to 20. Two network messages are needed to fetch the segmented strings.

Table 4−2 provides key information about the SQL_SGS_PREFETCH_ROWS parameter.

4.3.2.10 Specifying SQL_USERNAME and SQL_PASSWORD

The SQL_USERNAME and SQL_PASSWORD parameters specify the user name and password of a user to be authenticated for database access.

Table 4−2 provides key information about the SQL_USERNAME and SQL_PASSWORD parameters. See the Oracle Rdb7 Guide to SQL Programming for more information about the SQL_USERNAME and SQL_PASSWORD parameters.

4.3.2.11 Specifying SQL_TRANS_START_WAIT

The SQL_TRANS_START_WAIT parameter specifies the time in seconds that Oracle Rdb will wait when a new distributed transaction is started prior to an earlier one being ended. The default is three seconds. This delay comes into play only when a new distributed transaction is started while a previous one is still active. This allows Oracle Rdb to avoid a race condition caused by the fact that DECdtm might return control to an application from commit or rollback processing prior to notifying Oracle Rdb that the transaction should be ended. This may cause Oracle Rdb to report an inappropriate %RDB−E−EXCESS_TRANS error.

If your application is experiencing periodic %RDB−E−EXCESS_TRANS errors with distributed transactions and remote access even though the application is ending each transaction prior to starting a new one, it may be necessary to use the SQL_TRANS_START_WAIT parameter to extend the time Oracle Rdb waits prior to reporting an %RDB−E−EXCESS_TRANS error.

4.3.3 Modifying LOGIN.COM to Improve Network Performance

To improve performance over the network, modify login command files for server accounts on the remote node to allow faster processing. For example, if you define logical names for your databases, do so at the beginning of the LOGIN.COM file for the account Oracle Rdb will be running on the remote system. Then include the following command after the logical name definitions:

```
$ IF F$MODE() .EQS. "NETWORK" THEN $ EXIT)
```

4.4 Troubleshooting for Remote Access

The following sections describe some solutions to problems you may encounter while trying to attach to a remote database.

4.4.1 Retaining Asynchronous System Traps to Access a Remote Database

Using Oracle Rdb remotely requires the use of asynchronous system traps (ASTs) to send messages asynchronously. The remote interface is a client/server model. Each program issues an AST read on the network channel that connects the client and server. If a message is delivered by DECnet, the AST ensures that the message is handled immediately. If the message is a normal database message, a new AST is issued and the message that was received is processed normally.

The server is capable of serving multiple remote requests; this would not be possible with synchronous communication.

An Oracle Rdb routine never completes if ASTs are disabled and Oracle Rdb is attempting to access a database across DECnet. You should not disable ASTs when using Oracle Rdb.

4.4.2 Troubleshooting Application−Specific Problems

The following sections describe some solutions for application−specific problems. Not all problems or solutions are described here.

4.4.2.1 Avoiding Undetected Deadlock with Distributed Transactions

When you use distributed transactions to access databases on remote systems, undetected deadlocks may result. Deadlock occurs when two users are locking resources that both need, and neither user can continue until the other user ends a transaction. When deadlock occurs on the same node or the same cluster, the OpenVMS lock manager detects the deadlock and issues the deadlock error condition to one user. However, when a transaction accesses databases on remote systems, the OpenVMS lock manager cannot detect the deadlock. To help avoid distributed deadlock, Oracle Rdb provides the following methods to set the amount of time a transaction waits for locks to be released:

- The logical name RDM$BIND_LOCK_TIMEOUT_INTERVAL
- The WAIT interval clause of the SET TRANSACTION or DECLARE TRANSACTION statement

See the Oracle Rdb Guide to Distributed Transactions for more information.

4.4.2.2 Restrictions on Distributed Transactions Related to the DISTRIBTRAN Security Privilege

When you start a distributed transaction that uses a database on a remote node, Oracle Rdb checks that the account on the remote node has the DISTRIBTRAN privilege. For example, if you use a proxy account on the remote node, the proxy account must have the DISTRIBTRAN privilege on that database.

If you do not have the DISTRIBTRAN privilege and you try to start a distributed transaction, Oracle Rdb returns an error and does not start the transaction. This is especially important to remember when you use SQL. SQL starts a distributed transaction by default when you start a transaction that attaches to more than one database. The following privileges override the DISTRIBTRAN privilege:

- SQL privilege DBADM
- OpenVMS privilege SYSPRV
- OpenVMS privilege BYPASS

For more information about granting privileges, see the Oracle Rdb7 Guide to Database Design and Definition and the Oracle Rdb7 SQL Reference Manual.

4.4.3 Troubleshooting Summary

Table 4−4 shows some of the error messages you may encounter when trying to access a remote database. It does not show every possible problem that caused the error, nor does it show every possible solution. If you encounter an error not shown in Table 4−4, look in the RDB$REMOTE72 account directory for the NETSERVER.LOG file or, if you are using a proxy account, look in the top level directory of the user account for the NETSERVER.LOG file. This file displays more information about the errors you are encountering.

Table 4−4 Troubleshooting for Remote Access

Error	Problem	Solution
Error attaching to declared alias; Privilege denied by database facility	The RDBSERVER proxy access is not defined correctly	Using the NCP utility for DECnet Phase IV and the NCL utility for DECnet−Plus, define the proxy access for the RDBSERVER as incoming, outgoing, or both.
	There is no proxy account set up.	Set up a proxy account. See the Oracle Rdb7 Guide to SQL Programming.
	The database identifier [RDB$REMOTE] access is set to none or does not exist.	Grant the appropriate access to the identifier [RDB$REMOTE].
	The user application did not use a full file specification with user name and password to access the remote database.	Use a full file specification with user name and password or, specify the USER and USING clauses, which are required if the transport type is TCP/IP.
	The DBADM or DISTRIBTRAN privileges are not granted on all databases involved in a distributed	Grant the DBADM and DISTRIBTRAN privileges to the RDB$REMOTE72 account on all databases. See the OpenVMS documentation for more information.

	transaction.	
DECdtm is not installed on your system	DECdtm is not started on one or both of the nodes.	Start DECdtm on both nodes if the platform on both nodes is OpenVMS.
	The user application is trying to attach to more than one database, and SQL attempts to start a distributed transaction by default.	Define the logical name SQL$DISABLE_CONTEXT to be TRUE. See the Oracle Rdb Guide to Distributed Transactions.
Network object is unknown at remote node	DECdtm is not started on one or both of the nodes.	Start DECdtm on both nodes.
	The user application is trying to attach to more than one database, and SQL attempts to start a distributed transaction by default.	Define the logical name SQL$DISABLE_CONTEXT to be TRUE. See the Oracle Rdb Guide to Distributed Transactions.
	The RDBSERVER object is missing.	Run RDBSERVER_NCP.COM for DECnet Phase IV or RDBSERVER_NCL.COM for DECnet−Plus on the remote node.
Network partner aborted logical link	User application tried to access a remote database without a proxy account.	Set up a proxy account. See the Oracle Rdb7 Guide to SQL Programming.
	User application tried to access a remote database without using a full file specification with user name and password.	Use a full file specification with user name and password.
Error attaching to declared alias; Input or output error; Network partner exited	User application tried to access a remote database over the network. Commands in user's LOGIN.COM file may have redefined logical names.	Add this command to the beginning of your LOGIN.COM file on the remote system: *$ IF F$MODE() .EQS. "NETWORK" THEN $ EXIT*
Error attaching to declared schema; Input or output error; Login information invalid at	User application attempted to access a remote database	Disable the DISUSER flag in the RDB$REMOTE72 account.

remote node	while the RDB$REMOTE72 account was disabled.	
Does not reference a database known to Rdb; File not found	User application used the wrong file specification	Use the correct file specification.
	User application tried to attach to a remote database using a cluster alias	Use the actual node name and be sure each node has the RDBSERVER object proxy access set appropriately
No error returned; Process deadlocked	Two applications are trying to access the same resources at the same time, causing deadlock to occur.	Use the WAIT clause of the SET TRANSACTION statement or use the RDM$BIND_LOCK_TIMEOUT_INTERVAL logical name
%RDB−E−EXCESS_TRANS error even though prior transactions are committed or rolled back	DECdtm−induced race condition	Add the SQL_TRANS_START_WAIT parameter to the RDB$CLIENT_DEFAULTS file to specify a wait longer than three seconds.
Transaction log not found	DECdtm transaction log was not set up for one or both nodes	Use the LMCP utility to set up DECdtm transaction log. See the OpenVMS documentation for more information.

Appendix A
Sample Installation

This appendix contains a sample installation of an Oracle Rdb multiversion kit for OpenVMS I64. This installation is the initial installation of release 7.2 and was done on a system that had a prior version of Oracle Rdb already installed.

```
$ @SYS$UPDATE:VMSINSTAL RDBV72010IM072 DEV:[DIR]
    .
    .
    .
The following products will be processed:

  RDBV72010IM V7.2

        Beginning installation of RDBV72010IM V7.2 at 11:01

%VMSINSTAL−I−RESTORE, Restoring product save set A ...
%VMSINSTAL−I−RELMOVED, Product's release notes have been moved to
 SYS$HELP.
   Copyright © 1995, 2004, Oracle Corporation.  All Rights Reserved.

    The Rdb installation guide will be provided in SYS$HELP.

* Would you like to print the installation guide [NO]?

    *************************************************************

    SYSTEM MANAGER:

    The RMU Parallel Utilities require SQL/Services V7.0 or later.
    You currently do not have SQL/Services installed on your system.
    Please remember to install SQL/Services before you attempt to
    use this Rdb functionality.

    Please refer to the Oracle Rdb V7.2 Installation Guide and Release
    Notes for further information.

    *************************************************************

        Installation procedure for: "Oracle Rdb V7.2−010"

    You are about to install a multiversion Oracle Rdb kit.

    Be sure you have read the section entitled
        "Preparing Your System and the Installing Account"
    in the installation guide before continuing with the installation.

* Do you want to proceed [YES]?

    Checking system requirements ...


*************************************************************

This installation requires the creation of the RDB$REMOTE72
account.  The installation procedure will not proceed until you
enter a valid user identification code (UIC) for the RDB$REMOTE72
account.  The UIC must be unique.  Format [ggg,mmm].
```

```
    **************************************************************

* Enter UIC to be used for RDB$REMOTE72 account: [123,100]

    *************************************************************

    This installation requires the creation of the RDMAIJ72
    account.  The installation procedure will not proceed until you
    enter a valid user identification code (UIC) for the RDMAIJ72
    account.  The UIC must be unique.  Format [ggg,mmm].

    **************************************************************

* Enter UIC to be used for RDMAIJ72 account: [123,101]

    *************************************************************

    This installation requires the creation of the RDMSTT72
    account.  The installation procedure will not proceed until you
    enter a valid user identification code (UIC) for the RDMSTT72
    account.  The UIC must be unique.  Format [ggg,mmm].

    **************************************************************

* Enter UIC to be used for RDMSTT72 account: [123,102]
* Do you want to run the IVP after the installation [YES]?
* Do you want to purge files replaced by this installation [YES]?

    There are no more questions.

    Beginning installation ...

    Installing under VMS XADH-T3Z - 21-SEP-2004 11:03

%VMSINSTAL-I-RESTORE, Restoring product save set B ...
%VMSINSTAL-I-RESTORE, Restoring product save set C ...
%VMSINSTAL-I-RESTORE, Restoring product save set D ...
%VMSINSTAL-I-RESTORE, Restoring product save set E ...
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
 VMI$ROOT:[SYSHLP.EXAMPLES.RDB72].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
 VMI$ROOT:[SYSTEST.RDB72].
%VMSINSTAL-I-ACCOUNT, This installation creates an ACCOUNT named RDMSTT72.
%UAF-I-ADDMSG, user record successfully added
%UAF-I-RDBADDMSGU, identifier RDMSTT72 value [000123,000102] added to rights
 database
%VMSINSTAL-I-ACCOUNT, This installation creates an ACCOUNT named RDMSTT72.
%UAF-I-ADDMSG, user record successfully added
%UAF-I-RDBADDMSGU, identifier RDMSTT72 value [000123,000102] added to rights
database
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named RDMSTT72.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
VMI$ROOT:[RDMSTT72].
        *************************************************************

        Oracle Trace has not been installed. Now storing the RDBVMS
        facility definition into SYS$SHARE:EPC$FACILITY.TLB.
        After installing Oracle Trace, the facility definition
        may be placed in the Oracle Trace administration database
        Please refer to the Oracle Trace User's guide for instructions
        on how to insert binary facility definitions into the
        Oracle Trace administration database.

        *************************************************************

        *************************************************************

        SYSTEM MANAGER:

        The following command line MUST be added to the system startup
        command file SYS$STARTUP:SYSTARTUP_VMS.COM for all nodes that will
        be running Oracle Rdb.

            $ @SYS$STARTUP:RMONSTART72

        The following command line should be added to the system
        shutdown command file SYS$MANAGER:SYSHUTDWN.COM for all nodes
        that will be running Oracle Rdb.

            $ @SYS$MANAGER:RMONSTOP72

        *************************************************************

    %VMSINSTAL−I−ACCOUNT, This installation creates an ACCOUNT named
    RDB$REMOTE72.
    %UAF−I−ADDMSG, user record successfully added
    %UAF−I−RDBADDMSGU, identifier RDB$REMOTE72 value [000123,000100] added
    to rights database
    %VMSINSTAL−I−ACCOUNT, This installation updates an ACCOUNT named
    RDB$REMOTE72.
    %UAF−I−MDFYMSG, user record(s) updated

    *************************************************************

    SYSTEM MANAGER:

    To have remote access on another node which shares
    this cluster common root directory, you must configure the
    node's DECnet to recognize RDBSERVER.  Do the following:

        a) Log in to that node.
        b) $ @SYS$COMMON:[SYS$STARTUP]RDBSERVER_NCP.COM.

    This command procedure configures RDBSERVER with DECnet on
    that node.  This procedure needs to be executed only ONCE
    per node.

    *************************************************************

%VMSINSTAL−I−SYSDIR, This product creates system disk directory
VMI$ROOT:[RDB$REMOTE72].

%VMSINSTAL−I−ACCOUNT, This installation creates an ACCOUNT named RDMAIJ72.
%UAF−I−ADDMSG, user record successfully added
%UAF−I−RDBADDMSGU, identifier RDMAIJ72 value [000123,000101] added to rights
database
%VMSINSTAL−I−ACCOUNT, This installation updates an ACCOUNT named RDMAIJ72.
%UAF−I−MDFYMSG, user record(s) updated
%VMSINSTAL−I−SYSDIR, This product creates system disk directory
VMI$ROOT:[RDMAIJ72].

    *************************************************************
```

```
   SQL has been provided with Language−Sensitive Editor(LSE)
   support using the VMS LSE language.

   *************************************************************

   *************************************************************

   The Oracle Rdb Installation Verification Procedure (IVP) has
   been provided in SYS$COMMON:[SYSTEST].

   It is invoked using the commands:

     $ SET DEFAULT SYS$COMMON:[SYSTEST]
     $ @RDB$IVP72

   *************************************************************

   *************************************************************

   The release notes for Oracle Rdb are available in the file
   SYS$HELP:RDB072.RELEASE_NOTES

   *************************************************************

   %VMSINSTAL−I−MOVEFILES, Files will now be moved to their target
   directories...
Current PROCESS Oracle Rdb environment is version V7.2−010 (MULTIVERSION)
Current PROCESS SQL environment is version V7.2−010 (MULTIVERSION)
Current PROCESS Rdb/Dispatch environment is version V7.2−010 (MULTIVERSION)
Oracle Rdb monitor (RDMS_MONITOR72) started

Executing IVP for: Oracle Rdb V7.2−010

Current PROCESS Oracle Rdb environment is version V7.2−010 (MULTIVERSION)
Current PROCESS SQL environment is version V7.2−010 (MULTIVERSION)
Current PROCESS Rdb/Dispatch environment is version V7.2−010 (MULTIVERSION)
    Copyright © 1995, 2004, Oracle Corporation.  All Rights Reserved.

   Building the test database.

   Beginning Installation Verification Tests.

   Running the after−image journaling test.
       Test completed successfully

   Running the RDBPRE/BASIC precompiler test.
       Test completed successfully

   Running the RDBPRE/COBOL precompiler test.
       Test completed successfully

   Running the RDBPRE/FORTRAN precompiler test.
       Test completed successfully

   Running the RDML/DEC C preprocessor test.
       Test completed successfully

   Running the RDML/PASCAL preprocessor test.
       Test completed successfully

   Restoring the SQL database.
```

```
    Restore completed successfully

Running the Interactive SQL test.
    Test completed successfully

Running the Dynamic SQL test.
    Test completed successfully

Running the COBOL precompiler test.
    Test completed successfully

Running the FORTRAN precompiler test.
    Test completed successfully

Running the DEC C precompiler test.
    Test completed successfully

Running the PASCAL precompiler test.
    Test completed successfully

Running the SQL MODULE LANGUAGE test for BASIC.
    Test completed successfully

Running the SQL MODULE LANGUAGE test for C.
    Test completed successfully

IVP completed for: Oracle Rdb V7.2−010

    Installation of RDBV72010IM V7.2 completed at 11:09

Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

Creating installation data file: VMI$ROOT:[SYSUPD]RDBV72010IM072.VMI_DATA

    VMSINSTAL procedure done at 11:09
```

Appendix B
OpenVMS Security and Oracle Rdb

This appendix discusses the use of OpenVMS security features by Oracle Rdb.

B.1 OpenVMS Privileges Used to Install Oracle Rdb

Oracle Rdb must be installed from a privileged account. Usually, the SYSTEM account is used. The VMSINSTAL procedure is located in SYS$UPDATE, which is a restricted directory. The OpenVMS SETPRV privilege is required to run VMSINSTAL. The VMSINSTAL procedure then grants all privileges other than BYPASS. (The VMSINSTAL procedure also turns off BYPASS at the start of the installation.)

B.2 OpenVMS Privileges Required for Oracle RMU Commands

An Oracle Rdb database is protected by a combination of Oracle Rdb, Oracle RMU, and OpenVMS privileges. OpenVMS privileges are not necessary to use data manipulation or data definition statements. Oracle RMU privileges are used to control access to most database maintenance operations (for more information on Oracle RMU privileges, see the Oracle Rdb Release Notes and the Oracle Rdb7 Oracle RMU Reference Manual). However, some database maintenance operations still require OpenVMS privileges. Table B−1 lists the maintenance operations and indicates the required OpenVMS privilege.

Table B−1 Security Controls Required to Use Oracle RMU Functions

Oracle RMU Function	OpenVMS Privilege
Start database monitor	SETPRV
Reopen database monitor log	WORLD
Stop database monitor	WORLD
Show locks on databases	WORLD
Show databases in use	WORLD

Note

Start the monitor from the SYSTEM account that has the SETPRV privilege. The process starting the monitor attempts to give the monitor all privileges; the privileges required are as follows: ALTPRI, CMKRNL, DETACH, PSWAPM, SETPRV, SYSGBL, SYSNAM, and WORLD.

Oracle RMU functions require OpenVMS privileges when the function:

- Operates across multiple databases (such as the monitor−related commands)
- Does not operate on any database (such as the Oracle RMU Show command with the System qualifier)

B.3 OpenVMS Privileges That Override Oracle Rdb Protection

Certain OpenVMS privileges can override Oracle Rdb protection. Therefore, you must be very careful assigning OpenVMS privileges. The distinction between Oracle Rdb and OpenVMS privileges is that OpenVMS privileges are systemwide, while Oracle Rdb privileges are associated with a particular database or database object. Table B−2 indicates which Oracle Rdb privileges can be bypassed by users possessing certain OpenVMS privileges.

Table B−2 OpenVMS Privileges That Override Oracle Rdb Privileges

OpenVMS Privilege	Overridden Oracle Rdb Privileges
BYPASS	All privileges except DBADM, SECURITY, and DBCTR
READALL	SELECT database or table privilege
SYSPRV	All privileges except SECURITY
OPER	SELECT database privilege
SECURITY	SELECT database privilege, SECURITY database privilege, and DBCTRL

The Oracle Rdb7 Guide to Database Design and Definition includes a table indicating which actions can be performed with which OpenVMS and Oracle Rdb privileges.

Note

Certain sites might want to restrict the ability of users to create their own databases. These sites would have to define the RDBVMS$CREATE_DB logical name. When you use this logical name, other installed Oracle and third−party products will not be able to use Oracle Rdb to create Oracle Rdb databases. Therefore, you must deassign this logical name whenever users of such products need to create an Oracle Rdb database. More information on the use of this logical name can be found in the Oracle Rdb7 Guide to Database Design and Definition.

B.4 OpenVMS Protection of Oracle Rdb Files

Oracle Rdb sets the following OpenVMS default protection for all database files:
SYSTEM:READ,WRITE,EXECUTE,DELETE; OWNER:READ,WRITE; GROUP: , WORLD:

This affects the following files:

- Database root (.RDB) and its associated ACL
- Recovery−unit journal (.RUJ)
- After−image journal (.AIJ)
- Snapshot (.SNP)
- Storage area (.RDA)

These restrictions protect the database from applications or processes not using Oracle Rdb. Oracle Rdb uses the OpenVMS SYSPRV privilege to open database files, then checks that user's user identification code (UIC) against the Oracle Rdb access privilege set to determine access to database objects. Section B.5 discusses protection specific to Oracle Rdb.

B.5 Oracle Rdb Internal Protection

Internal Oracle Rdb protection depends on the use of access privilege sets (APSs) that connect database subjects (users) and objects with certain privileges. Oracle Rdb uses the standard OpenVMS identifiers to identify database subjects.

The UIC of the process owner is used by Oracle Rdb to identify the individual who is accessing the database. No separate user identifiers are supported by Oracle Rdb, and no separate authentication of users is performed.

Database administrators can choose between ACL−style and ANSI/ISO−style protection when using the SQL interface to Oracle Rdb. In ACL−style protection, three types of OpenVMS identifiers can be used:

- User identification codes (UICs)
  The following are all valid UICs:

  ```
  [SYSTEMS,JONES]
  K_JONES
  [354,567]
  [250,*]
  ```
- General identifiers that specify a user or set of users
  For example:

  ```
  DATAENTRY
  PROGRAMMERS
  MANAGERS
  SECRETARIES
  ```
- System−defined identifiers
  For example:

  ```
  BATCH
  NETWORK
  INTERACTIVE
  LOCAL
  DIALUP
  REMOTE
  ```

Each identifier is associated with a set of access privileges to specify which operations that user or user group can perform on the database or database table, view, or column. In ANSI/ISO−style protection, only a specific UIC can be used. Wildcards are permitted only to specify public access, as in [*,*].

Database objects (database, table, view, or column) are associated with an APS that indicates which operations certain users can perform on that object. The owner or creator of a database owns the database files and has the ability to grant or revoke privileges for that database's subjects and objects.

For more information on other aspects of Oracle Rdb security, see the Oracle Rdb7 Guide to Database Design and Definition.

B.6 Auditing

Oracle Rdb employs a security auditing system that closely models that of the OpenVMS system.

A database is maintained that describes the Oracle Rdb audit events that are enabled. Such events are enabled on a per database basis so that each database can be audited differently. Oracle RMU includes RMU Set Audit and RMU Show Audit commands to modify and display the event auditing characteristics. As with the OpenVMS system, Oracle Rdb has its own audit analysis command (RMU Load command with the Audit qualifier) to assist in reviewing the audit trail.

To accomplish security auditing, Oracle Rdb communicates with the OpenVMS AUDIT_SERVER process, which stores security audit records in the security audit journal and relays security alarm messages to the appropriate display process. Thus, Oracle Rdb audit information can coexist with OpenVMS audit information so that all system audit records can be retrieved from one location by the OpenVMS security administrator using a single OpenVMS audit analysis tool.

For more information on Oracle Rdb auditing capabilities, see the Oracle Rdb7 Guide to Database Maintenance. For more information on OpenVMS auditing capabilities, see the OpenVMS documentation set.

| Contents