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 7.3.3.2

November 2019

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 7.3.3.2 on the HPE OpenVMS for Alpha and HPE OpenVMS Industry Standard 64 for Integrity Servers operating systems and OpenVMS V8.4–1H1 or later operating systems from VMS Software Inc. (VSI). You do not have to install a previous version of Oracle Rdb before installing Oracle Rdb Release 7.3.3.2.

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 7.3.2.3 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 HPE 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 7.3.3.2 of Oracle Rdb software is often referred to as Release 7.3.3.2 or V7.3.3.2.
- 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 7.3.3.2, V7.3.3.2 and release 7.3.3.2.
- 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 Enterprise is referred to as HPE.
- VMS Software Inc is referred to as VSI.
- DECnet and DECnet−Plus refer respectively to HPE DECnet for OpenVMS and HPE DECnet−Plus for OpenVMS. DECnet Phase IV is used interchangeably with DECnet in this document.

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
  - ♦ Hot Standby (replicate an Oracle Rdb database at a remote standby site)
  - ♦ Support for the execution of previously developed applications
- Programmer for Rdb (Rdb Compilers) includes all Oracle Rdb compilers, including, for example:
  - ♦ SQL precompiler
  - ♦ SQL Module Language processor
  - ♦ RDBPRE precompiler
  - ♦ RDML precompiler

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
  - ♦ 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:

```
https://www.oracle.com/database/technologies/related/rdb.html
```

1.2.1 Operating System Requirements

Oracle Rdb Release 7.3.3.2 requires one of the following OpenVMS environments:

- OpenVMS Alpha version 8.4 or later
- OpenVMS I64 version 8.4 or later
- To run on i6 machines, VMS 8.4–2L1 or later is required

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 (RMONSTART73.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 7.3.3.2 is compatible with many Oracle software products. These products include Oracle CDD/Repository, Oracle Trace, Oracle Replication Option for Rdb, Oracle JDBC for Rdb and Oracle Rdb Connectivity Manager.

Oracle Rdb Release 7.3.3.2 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 recommends using Oracle CDD/Repository Release 7.2.0.6 or later with Oracle Rdb Release 7.3.3.2. Use the Common Dictionary Operator (CDO) utility to see the version of Oracle CDD/Repository currently installed on your system.

  ```
  $ REPOSITORY OPERATOR SHOW VERSION
  Installed version of Oracle CDD/Repository is V7.2-060
  ```

  See Section 1.9.3 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 7.3.3.2 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 Connectivity Manager (ORCM)
  Oracle Rdb Connectivity Manager (ORCM) is a graphical user interface (GUI) that allows remote management of SQL/Services and JDBC servers as well as providing a convenient interface to other Oracle Rdb management tasks.
  ORCM provides the following:
    ♦ Remote Server management of SQL/Services and JDBC servers
    ♦ A Database explorer to view database metadata
    ♦ Various RMU tools for remote management of databases
    ♦ The ability to create and execute user defined External tools to carry out RMU and DCL operations on remote servers and databases
    ♦ Event notification and monitoring of events raised by JDBC servers and user defined events
    ♦ A Database Schema difference tool to view the differences between two database schemas
    ♦ An RMU Statistics tool providing statistical displays similar to RMU/SHOW statistics
    ♦ An Oracle Trace display tool allowing the playback and analysis of Oracle Trace logs
    ♦ A SQL Worksheet tool allowing the execution of SQL statements on remote Oracle Rdb databases

ORCM is a separate installation from Oracle Rdb and may be installed and run from any platform where Java is available. See the Oracle Rdb Connectivity Manager Release Notes for additional information.

Oracle Rdb Release 7.3.3.2 is compatible with many software products from HPE, including COBOL, ACMS and HPE DATATRIEVE.

1.4 Hardware Note on Alpha EV56 Requirement

Oracle Rdb has been optimized for the Alpha EV56 platform as a minimum. If running on an older platform, the maximum Rdb version you can run is Rdb Release 7.2.5.7.

1.5 Disk Space Requirements

The minimum storage requirement for installing Oracle Rdb for OpenVMS Alpha is 350,000 blocks; the minimum storage requirement for installing Oracle Rdb for OpenVMS I64 is 600,000 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.

1.6 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
WSDEFAULT	150
WSQUOTA	512

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

1.7 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

1.8 Database Server Process Priority Clarification

By default, the database servers (ABS, ALS, DBR, LCS, LRS, RCS) created by the Rdb monitor inherit their VMS process scheduling base priority from the Rdb monitor process. The default priority for the Rdb monitor process is 15.

Individual server priorities can be explicitly controlled via system–wide logical names as described in Table 1–4.

Table 1–4 Server Process Priority Logical Names

Logical Name	Use
RDM$BIND_ABS_PRIORITY	Base Priority for the ABS Server process
RDM$BIND_ALS_PRIORITY	Base Priority for the ALS Server process
RDM$BIND_DBR_PRIORITY	Base Priority for the DBR Server process
RDM$BIND_LCS_PRIORITY	Base Priority for the LCS Server process
RDM$BIND_LRS_PRIORITY	Base Priority for the LRS Server process
RDM$BIND_RCS_PRIORITY	Base Priority for the RCS Server process

The RDMAIJSERVER account for Hot Standby is created specifying an account priority of 15. The priority of AIJ server processes on your system can be restricted with the system–wide logical name RDM$BIND_AIJSRV_PRIORITY. If this logical name is defined to a value less than 15, an AIJ server process will adjust its base priority to the value specified when the AIJ server process starts. Values from 0 to 31 are allowed for RDM$BIND_AIJSRV_PRIORITY, but the process is not able to raise its priority above the RDMAIJSERVER account value.

For most applications and systems, Oracle discourages changing the server process priorities.

1.9 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 7.3.3.2.

1.9.1 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 7.3.3.2.

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 1.9.1.1 describes how to upgrade from Oracle Rdb Release 7.0, 7.1 or 7.2 to Release 7.3.3.2 using the RMU Convert command.

You cannot convert databases earlier than Oracle Rdb Release 7.0 directly to Release 7.3.3.2.

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 1.9.1.1 for instructions to convert the intermediate database to Release 7.3.3.2.

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 1.9.1.1 for instructions to convert the intermediate database to release 7.3.3.2.

1.9.1.1 Upgrading from an Oracle Rdb Prior Release to Release 7.3.3.2

If you are using a version of Oracle Rdb from release 7.0, 7.1 or 7.2 and want to upgrade to Release 7.3.3.2, 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 7.3.3.2. After installing:
   a. Reset the DCL tables on each node of the cluster.
   b. Start the Oracle Rdb monitor process by executing RMONSTART73.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.

1.9.2 Reverting to Release 7.0, 7.1 or 7.2 from Release 7.3.3.2

If you have converted a prior version database to release 7.3.3.2 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 7.3.3.2 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, 7.1 or 7.2 and install Release 7.0, 7.1 or 7.2 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 7.3.3.2. These files are no longer useful after you have made a backup of the reverted database.

1.9.3 CDD/Repository Considerations

You must install Oracle Rdb before installing CDD/Repository. If you are also installing Oracle CODASYL DBMS, the order of installation is Oracle Rdb first, then CDD/Repository, and finally Oracle CODASYL DBMS.

1.9.4 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.

1.9.5 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
```

1.9.6 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.

1.9.7 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 1−5 lists the system parameter values required for installing Oracle Rdb.

Table 1−5 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 1−5 as 512−byte pagelets, which are not CPU−specific. GBLPAGFIL values on OpenVMS systems are expressed in CPU−specific pages, typically 8192 bytes.

Table 1−5 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.

1.9.7.1 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 1.9.7.3 describes the procedures for changing system parameter values.

1.9.7.2 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 1–5. 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 1.9.7.3 describes the procedures for changing system parameter values.

1.9.7.3 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.

1.9.7.4 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 1–5.

1.9.8 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.

1.9.9 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.

1.9.10 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.

1.9.11 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 HPE 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 1.5
Back up existing databases.	See Section 1.9.1
Resolve repository considerations.	See Section 1.9.3
Confirm adequate account privileges.	See Section 1.9.4
Confirm account password is unlocked.	See Section 1.9.5
Confirm adequate account quotas.	See Section 1.9.6
Confirm system parameter values.	See Section 1.9.7
Back up your system disk.	See Section 1.9.8
Disable access to online help.	See Section 1.9.9
Prevent access to the system.	See Section 1.9.10

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$HELP:RDB073xx.RELEASE_NOTES.
- The file specification for the PostScript format is SYS$HELP:RDB073xx_RELEASE_NOTES.PS.
- The file specification for the PDF format is SYS$HELP:RDB073xx_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.

Note

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

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:
    - *RDBV73320IM073* for OpenVMS I64
    - *RDBV73320AM073* 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:[RDBV73320IM073]
      ```
    - 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
   The installation procedure will ask where the installation files are to be found.

   ```
   * Where will the distribution volumes be mounted: ryerox::rdb_rel73:[rdbv73320ki
   t_imv]

   Enter the products to be processed from the first distribution volume set.
   * Products: RDBV73320IM073
   * Enter installation options you wish to use (none):
   The following products will be processed:

     RDBV73320IM V7.3
   ```

```
       Beginning installation of RDBV73320IM V7.3 at 08:10
```

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

```
No signature manifests found for RDBV73320IM073

* Do you want to install this product [NO]? yes

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
   Copyright © 1995, 2019, Oracle Corporation.  All Rights Reserved.

        Installation procedure for: "Oracle Rdb V7.3-320"

    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 ...
```

3. Entering a UIC for the RDB$REMOTE73 account
   If this is the initial installation of Oracle Rdb, the procedure creates an account called
   RDB$REMOTE73. You must choose a unique user identification code (UIC) for this account, which
   the installation procedure uses when it creates the RDB$REMOTE73 account. The installation
   procedure prompts you to enter a UIC.

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

   This installation requires the creation of the RDB$REMOTE73
   account.  The installation procedure will not proceed until you
   enter a valid user identification code (UIC) for the RDB$REMOTE73
   account.  The UIC must be unique.  Format [ggg,mmm].

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

 * Enter UIC to be used for RDB$REMOTE73 account: [250,201]
```

4. Creating the RDMAIJ73 account
   The installation procedure requires the creation of the RDMAIJ73 account. You must choose a unique
   user identification code (UIC) for this account, which the installation procedure uses when it creates
   the RDMAIJ73 account. The installation procedure prompts you to enter a UIC.

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

   This installation requires the creation of the RDMAIJ73
   account.  The installation procedure will not proceed until you
   enter a valid user identification code (UIC) for the RDMAIJ73
   account.  The UIC must be unique.  Format [ggg,mmm].

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

 * Enter UIC to be used for RDMAIJ73 account: [250,202]
```

5. Creating the RDMSTT73 account
   The installation procedure requires the creation of the RDMSTT73 account. You must choose a
   unique user identification code (UIC) for this account, which the installation procedure uses when it

creates the RDMSTT73 account. The installation procedure prompts you to enter a UIC.

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

    This installation requires the creation of the RDMSTT73
    account.  The installation procedure will not proceed until you
    enter a valid user identification code (UIC) for the RDMSTT73
    account.  The UIC must be unique.  Format [ggg,mmm].

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

```
  * Enter UIC to be used for RDMSTT73 account: [250,203]
```

6. 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 3.12.

7. 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]?
```

8. 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 ...

    OpenVMS IA64 V8.4-2L1 - 27-SEP-2019 8:12

%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 ...
    .
    .
    .
```

9. Additional informational messages appear in releases of Oracle Rdb after 7.3.3.1 (ie, starting in Rdb 7.3.3.2) that show that the product installation has been added to the PCSI product history to give better data on tracking what products/versions have been installed.

```
The following product has been selected:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)
```

```
The following product will be registered:
    ORCL I64VMS RDB73 V7.3-320              DISK$CLYPPR84_2:[VMS$COMMON.]


File lookup pass starting ...


Portion done: 0%
...100%

File lookup pass completed search for all files listed in the product's PDF
Total files searched: 0   Files present: 0   Files absent: 0

The following product has been registered:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...
```

10. 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 V7.3-320
```

11. Completing the installation

The following messages indicate that the entire installation procedure is complete:

```
 Installation of RDBV73320IM v7.3 completed at 8:15

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

   Creating installation data file: VMI$ROOT:[SYSUPD]RDBV73320IM073.VMI_DATA

Enter the products to be processed from the next distribution volume set.
* Products:
      VMSINSTAL procedure done at 8:16
```


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 7.3.3.2 is named RDMSHR73.EXE; the image for interactive SQL is named SQL$73.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 in an Oracle Rdb multiversion environment, take some or all of the following steps, depending on the combination of Oracle Rdb versions:

- To install Oracle CDD/Repository in an Oracle Rdb Release 7.3 environment, use the RDB$SETVER command procedure, described in Section 2.4.1 to set up Oracle Rdb Release 7.3 as the active version during the installation. Failure to do this causes the Oracle CDD/Repository IVP procedure to fail.
- Oracle CDD/Repository supports Oracle Rdb multiversioning. If you install Oracle CDD/Repository and you also want to create a repository for a particular version of Oracle Rdb, (for example, Oracle Rdb Release 7.2), execute these steps:
    1. Use the RDB$SETVER command procedure to set up the Oracle Rdb Release 7.2 environment.

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

       ```
       $ 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 7.3.3.2 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 7.3.3.2 logical names, execute the following command:

```
$ @SYS$LIBRARY:RDB$SETVER 7.3
```

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 7.3.3.2 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 7.3.3.2 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 7.3 /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 7.3 /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): 7.3
Current PROCESS Oracle Rdb environment is version V7.3−32 (MULTIVERSION)
Current PROCESS SQL environment is version V7.3−32 (MULTIVERSION)
Current PROCESS Rdb/Dispatch environment is version V7.3−32 (MULTIVERSION)
```

The previous example sets the multiversion variant of Oracle Rdb Release 7.3.3.2 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 7.3 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 7.3 /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 V7.3-32 (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" = "73" (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 7.3, 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$SETVER.COM command procedure sets logical names and symbols for most Oracle Rdb images to point to variated file names. Thus, the symbol SQL$ points to SQL$73.EXE, and SQL$PRE points to SQL$PRE73.EXE, if you have set the version to 7.3. The following examples show how to determine your Oracle Rdb environment:

```
$ RMU/SHOW VERSION
Executing RMU for Oracle Rdb V7.3-32
$ RUN SQL$
SQL> ATTACH 'FILENAME PERSONNEL';
SQL> SHOW VERSIONS
Current version of SQL is: SQL V7.3-32
Underlying versions are:
    Database with filename PERSONNEL
```

```
        Oracle Rdb V7.3-32
        Rdb/Dispatch V7.3-32
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 V7.3
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 7.2 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 72.0 is not compatible with software
version 73.0
SQL> SHOW VERSION
Current version of SQL is: SQL V7.3-32
```

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$SHOVER command procedure. Previously, these products usually read the version number from the header of one of the standard Oracle Rdb images, such as RDMSHRP. If you install the multiversion variant of Oracle Rdb Release 7.3.3.2, the old image names may not be available to determine the version number. The RDB$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$INSTALLED_VERSIONS" = "V7.2-0" (LNM$PROCESS_TABLE)
        = "*V7.3-0"
```

In this example, V7.2–0 indicates that Oracle Rdb Release 7.2 is installed; *V7.3–0 indicates that the multiversion variant of Oracle Rdb Release 7.3.3.2 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, 7.3, the logical names show only the information of the version indicated.

```
$ @SYS$LIBRARY:RDB$SHOVER.COM VERSIONS 7.3
   "RDBVMS$INSTALLED_VERSIONS" = V7.3-32 LNM$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"
        = "*V7.3-0"
   "SQL$INSTALLED_VERSIONS" = "V7.1-0" (LNM$PROCESS_TABLE)
        = "*V7.2-0"
        = "*V7.3-0"
   "RDB$DISPATCH_INSTALLED_VERSIONS" = "V7.1-0" (LNM$PROCESS_TABLE)
        = "*V7.2-0"
        = "*V7.3-0"
```

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 7.3 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 7.3, SQL$USER translates to SQL$USER73.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 7.3.3.2 are located in SYS$LIBRARY:LSE$SQL73MV_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$LIBRARY:LSE$SQL73MV_ENVIRONMENT.ENV TEST.SQL
  $ LSEDIT /SYSTEM_ENVIRONMENT=SYS$LIBRARY:LSE$SQL73MV_ENVIRONMENT.ENV TEST.SQLMOD
  ```
- ♦ Define a process logical name first and invoke LSEDIT without a qualifier.

  ```
  $ DEFINE LSE$SYSTEM_ENVIRONMENT SYS$LIBRARY:LSE$SQL73MV_ENVIRONMENT.ENV
  $ LSEDIT TEST.SQL
  $ LSEDIT TEST.SQLMOD
  ```

2.4.7 Accessing Remote Databases

You can access a release 7.3.3.2 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$REMOTE73 account defines the appropriate RDBSERVER and RDB$SHARE images to run. Users must specify the RDB$REMOTE73 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$REMOTE73 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$REMOTE73 account, you must add the following lines to the account's LOGIN.COM file:

```
$ DEFINE RDBSERVER SYS$COMMON:RDBSERVER73.EXE
$ DEFINE RDMS$VERSION_VARIANT 73
```

2.4.7.2 Using TCP/IP Transport

You can define your own TCP/IP 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 V7.2 database you can create a TCP/IP service called rdbsrv72 that uses the user name rdb$remote72. Then add the following line to your client configuration file to use the TCP/IP service:

```
SQL_ALTERNATE_SERVICE_NAME rdbsrv72
```

For more details on how to set up TCP/IP services, see Section 4.2.2. For more information about configuration files, see Section 4.4.1 and Section 4.4.2.

In this example, if you choose to use a different user name than rdb$remote72 to access a V7.2 database, the LOGIN.COM file of that user must contain the following lines:

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

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 7.3.3.2 help on any of the varianted topics in the previous list, type HELP and the topic name with a "73" suffix:

```
$ HELP SQL73
```

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$SHARE73.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 72, RDBSHR.EXE calls RDB$SHARE72.EXE, which contains the release 7.2 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 V7.3.3.2 has failed.
```

If the IVP fails, you see these messages:

```
The RDB V7.3.3.2 Installation Verification Procedure failed.

%VMSINSTAL-E-IVPFAIL, The IVP for RDB V7.3.3.2 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 1.9.4.
♦ 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 1.5 for disk space requirements.
♦ Insufficient system parameter values
  You must have the necessary minimum settings for system parameters. See Section 1.9.7.
♦ Insufficient quotas for successful installation
  You must have the necessary minimum account quotas set. See Section 3.6.
♦ OpenVMS Help Library currently in use
♦ RMONSTART73.COM procedure found in SYS$SPECIFIC:[SYS$STARTUP]
  The IVP will fail if it executes an old version of the RMONSTART73.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 7.3.3.2 Media

The Oracle Rdb Release 7.3.3.2 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:

♦ JRDBV73320AM073 for OpenVMS Alpha operating systems
♦ JRDBV73320IM073 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 3.8
Enable SQL SET LANGUAGE (optional).	See Section 3.9.1
Enable Oracle Trace support (optional).	See Section 3.9.2
Enable RDB$REMOTE73 account (optional).	See Section 3.9.3
Install images as resident on OpenVMS (optional).	See Section 3.10
Start Oracle CDD/Repository (optional).	See Section 3.11
Run the Installation Verification Procedure (IVP) (optional).	See Section 3.12
Reset read−only storage areas.	See Section 3.13
Delete previous versions of Oracle Rdb.	See Section 3.14

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 1.9.7.4 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:RMONSTART73.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 RMONSTART73.COM starts the multiversion variant of Oracle Rdb Release 7.3.3.2.

```
$ @SYS$MANAGER:STARTNET.COM
       .
       .
       .
$ @SYS$STARTUP:RMONSTART.COM
$ @SYS$STARTUP:RMONSTART73.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 RMONSTART73.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:RMONSTOP73.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:RMONSTOP73.COM
```

To invoke the RMONSTOP73.COM command procedure, you need the user privilege SETPRV or the privileges CMKRNL, SYSNAM, and WORLD. The RMONSTOP73.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$LIBRARY:SQL$USER73.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 1.9.7. 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 RMONSTART73.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, RMONSTART73.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$MANAGER:RDBSERVER_NCP.COM. Note that this command procedure is called and runs from RMONSTART73.COM. See Section 3.5.2 for more information on RDBSERVER_NCP.COM.
   ◊ For DECnet–Plus environments, run SYS$MANAGER:RDBSERVER_NCL.COM. This command procedure is called and runs from RMONSTART73.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:RMONSTART73
SYSMAN> DO @SYS$TEST:RDB$IVP73
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 RMONSTART73.COM. If you execute RMONSTART73.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. RMONSTART73.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 RMONSTART73.COM. If you execute RMONSTART73.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 RMONSTART73.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

A user must have, at a minimum, the OpenVMS privileges NETMBX and TMPMBX to work correctly with Oracle Rdb.

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
ASTLM	The larger of 100 or the sum of the following: • 5 (for Sort work, AIJ and RUJ I/O operations) • Database asynchronous batch write buffer count times the database buffer size • Database asynchronous prefetch buffer count times the database buffer size
BIOLM	16
BYTLM	The larger of 1,048,576 or 512 times the sum of the following: • 1024 (for Sort work, AIJ and RUJ I/O operations) • Database asynchronous batch write buffer count times the database buffer size • Database asynchronous prefetch buffer count times the database buffer size • Number of database storage areas, snapshot storage areas and AIJ files
DIOLM	150 Larger values combined with high performance storage subsystems and large asynchronous I/O counts can allow increased throughput.
ENQLM	5000 Oracle recommends a value of 32767 in the UAF account entry which enables a virtually unlimited number of database−related locks held
FILLM	25 more than the total number of database storage areas, snapshot storage areas, and after image journals
PGFLQUOTA	700000 Large enough to contain the process's program use of buffers and code along with Rdb's use of buffers and code.
TQELM	100
WSQUOTA, WSEXTENT	Large enough to avoid excessive process page faulting

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:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> SHOW SMITH
```

To change quotas and privileges, use the MODIFY command:

```
UAF> MODIFY account−name /quota-name=NNN /PRIVILEGE=(privs) /DEFPRIV=(privs)
```

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 Typical Process Quotas for Application Environments

System parameters and process quotas described in the previous sections are minimum values, and each application environment may require different values. This section examines the quotas used by a typical Oracle Rdb production system.

The values suggested in this section are *suggested* settings; the settings required by users on your system might differ substantially. The suggested values are specific only to the use of Oracle Rdb. You should incorporate the values required for other OpenVMS layered products into the values you choose to use for Oracle Rdb and modify the values for each user as needed. If the OpenVMS defaults are greater than the values listed here, please use the OpenVMS defaults.

The UAF (user authorization file) parameters pertinent to Oracle Rdb are:

- ASTLM
  The asynchronous trap queue limit is a limit on the number of outstanding queued ASTs for a process. Use the following formula to determine a value for ASTLM:

  ```
  ASTLM > DIOLM + 32
  ```

  The AST queue limit is the maximum number of AST operations and scheduled wake−up requests that can be outstanding at any one time. The ASTLM value should be greater than the value for DIOLM, plus 32.
  By default, Oracle Rdb uses asynchronous I/O when possible to reduce the overall time to issue and wait for I/O. Setting ASTLM too low may limit the effectiveness of operations such as asynchronous prefetch and asynchronous batch write. A value of 5000 would not be unreasonable in an active production system.
- BIOLM
  Buffered I/O limit count is the maximum number of buffered I/O operations that can be outstanding at one time. Set this value at least to 150.
- BYTLM
  The buffered I/O byte limit is the maximum number of bytes of non−paged system dynamic memory that can be specified at one time by a user's job for transfer to outstanding buffered I/O operations. Set this to 1,048,576 or higher.
- DIOLM
  Direct I/O count limit is the maximum number of direct I/O operations (usually to disk) that can be outstanding at one time. Set this value to at least 150.
  In a busy, high I/O system environment, having this quota too low might limit throughput of your application. A value of 500 would not be unreasonable for such systems.
- ENQLM
  The lock queue limit sets the maximum number of locks that can be queued (held) at any one time by a process.
  However, your applications may need larger ENQLM values. Locks are required for each open storage area, each database page and possibly each record (which includes index nodes, hash buckets and data rows). The adjustable lock granularity (ALG) feature of Rdb attempts to lock a range of pages using a single lock, and only upon contention with other users, locks individual pages and records. Larger global buffer allocate sets or larger local buffer counts will require more locks be

acquired.

Complex transactions that lock many records will increase the number of locks required. As read/write transactions read and update rows they must be locked to preserve database integrity, and the pages and buffers holding those rows (or index nodes) will be locked to allow coordination with other updating processes.

Applications which attach to more than one database, or the same database multiple times, will also require sufficient ENQLM quota to cover all these database instances.

An ENQLM value of 32767 in a user's UAF record is treated as if there is no quota limit for that user. This means that the user is allowed to own up to 16,776,959 locks – the architectural maximum of the OpenVMS lock manager. This setting can greatly simplify process management in regards to locking. When an application uses this setting, the system manager should ensure that the OpenVMS system parameter RESHASHTBL is large enough to cope with the large number of resources. Use the MONITOR LOCK DCL command to assist in an appropriate setting.

- FILLM

The open file limit specifies the maximum number of files that a user's process can have open at a time. Set this to at least 1000.

Each database storage area (.rda), snapshot file (.snp), recovery unit file (.ruj), after image journal file (.aij), root file (.rdb), or log file can be opened by the user.

In addition to these database specific files, temporary (transient) work files might need to be opened by SORT, query execution, RMU/RECOVER, and so on.

For a multi−file database, set this value to the sum of the .rdb, .rda, .snp, .ruj, .aij, and the temporary work files for the application run by this process. Oracle recommends adding 25 to account for temporary files created during sorting and join operations, used by RMU operations, executable images and application data files.

If the FILLM quota is exceeded, the current operation aborts and an exceeded quota message is returned to the user. Increasing the FILLM quota requires a corresponding increase to the SYSGEN parameter CHANNELCNT if the new FILLM quota is higher than the current value of CHANNELCNT.

Note

Some RMU commands might need a larger FILLM quota. For instance, RMU/COPY_DATABASE will open each storage area, each storage area's snapshot file, and requires the same number of open files on the target copy database. This means at least 4 times the number of storage areas in the source database.

- PGFLQUOTA

Paging file limit is the maximum number of pages that the user's process can use in the system paging file.

The consumption of a process's virtual private pages (and, by definition, possible use of page file space) is related to number of database buffers, number of storage areas, tables, locks, query complexity and so on and is very application and configuration dependent. Oracle suggests that a value of 1,000,000 be considered as adequate for many database applications.

- PRCLM

The Subprocess creation limit is the maximum number of subprocesses that can exist at one time for the user's process. Oracle Rdb doesn't use subprocesses at runtime. However, in a development environment, the SQL Pre−compiler will start a subprocess to execute the host language compile command. So this value should be set to at least 2, plus any requirements of the application.

- TQELM

The timer creation limit specifies the number of timers that a process may have pending. Set to a minimum of 100.

- WSDEFAULT, WSQUOTA, WSEXTENT

  The default working set size, the working set quota, and the working set extent.

  WSDEFAULT sets the initial working set size limit for a user's process. WSQUOTA guarantees the user that the number of physical pages specified will be available. WSEXTENT sets the maximum number of pages on a system−wide basis for any working set.

  Set values higher for Rdb applications: WSDEFAULT 8192; WSQUOTA 16384 (depending on available memory); WSEXTENT 32767 or higher.

  The complexity of the application will impact the virtual memory requirements of the process. For example, query options such as *join*, *union distinct*, *group by* and *order by* will produce intermediate results that must be kept in memory before the final results are delivered to the application.

  If the database has enabled global buffers and has a large number of global buffers, you may need higher quotas for WSDEFAULT, WSQUOTA, and WSEXTENT. See the OpenVMS documentation set for a more detailed discussion on this topic and guidelines on setting initial working set values for tuning automatic working set adjustment parameters.

  If you perform any explicit sorting operations (ORDER BY, DISTINCT, GROUP BY, UNION DISTINCT) or an implicit sort operation (possibly as part of a join) and the query takes more time than expected to complete and you notice excessive disk access, check your WSEXTENT and WSQUOTA parameter values to see if they are set properly for the operation.

See the OpenVMS documentation set for information on how to use the AUTHORIZE utility to modify process quotas.

3.8 Converting Existing Databases

Users must use Oracle RMU to convert existing Oracle Rdb databases to a format compatible with Oracle Rdb Release 7.3 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 1.9.1 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 7.3.3.2. 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 7.3.3.2, perform the following steps:

1. Back up the prior version Oracle Rdb database, either before installing Rdb V7.3 or by using RDB$SETVER to set the environment to the correct version.

   ```
   $ @sys$share:rdb$setver 7.1
   Current PROCESS Oracle Rdb environment is version V7.1-521 (MULTIVERSION)
   Current PROCESS SQL environment is version V7.1-521 (MULTIVERSION)
   Current PROCESS Rdb/Dispatch environment is version X7.1-01 (MULTIVERSION)
   $
   $ rmu/backup MF_PERSONNEL mfp_backup
   $ rmu/backup/after MF_PERSONNEL mfp_aij_backup
   $
   $ @sys$share:rdb$setver 7.3
   Current PROCESS Oracle Rdb environment is version V7.3-320 (MULTIVERSION)
   Current PROCESS SQL environment is version V7.3-320 (MULTIVERSION)
   Current PROCESS Rdb/Dispatch environment is version X7.3-01 (MULTIVERSION)
   $
   ```

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, the newer version and the previous older version.
   The multiversion feature of Oracle Rdb enables you to test applications using the latest version of Oracle Rdb, while continuing to use other databases with the previous version of the software. However, you will not be able to perform data definition language (DDL) operations or use some new features 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
$ RMU/CONVERT/CONFIRM MF_PERSONNEL
%RMU-I-RMUTXT_000, Executing RMU for Oracle Rdb V7.3-320 on OpenVMS Alpha V8.4
Are you satisfied with the backup of USER$DISK1[TESTING]MF_PERSONNEL.RDB;1 and
any associated after image (.aij) files [N]? y
%RMU-I-AIJ_DISABLED, after-image journaling is being disabled temporarily for
the Convert operation
%RMU-I-LOGCONVRT, database root converted to current structure level
%RMU-S-CVTDBSUC, database USER$DISK1:[TESTING]MF_PERSONNEL.RDB;1 successfully
converted from version V7.1 to V7.3
%RMU-I-CVTCOMSUC, CONVERT committed for USER$DISK1:[TESTING]MF_PERSONNEL.RDB;1
to version V7.3
%RMU-I-LOGCREAIJ, created after-image journal file USER$DISK1:[TESTING]MFP.AIJ;2
%RMU-I-LOGMODSTR,    activated after-image journal "RDB$JOURNAL"
%RMU-W-DOFULLBCK, full database backup should be done to ensure future recovery
$
```

Users trying to access unconverted databases with release 7.3.3.2 software receive the following fatal
error messages:

```
SQL> attach 'file mf_personnel';
%SQL-F-ERRATTDEC, Error attaching to database mf_personnel
-RDB-F-WRONG_ODS, the on-disk structure of the database file is not supported
by this version
-RDMS-F-ROOTMAJVER, database format 70.1 is not compatible with software
version 73.0
```

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/CONFIRM MF_PERSONNEL
%RMU-I-RMUTXT_000, Executing RMU for Oracle Rdb V7.3-320 on OpenVMS Alpha V8.4
Are you satisfied with the backup of USER$DISK1[TESTING]MF_PERSONNEL.RDB;1 and
any associated after image (.aij) files [N]? N
$
```

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

```
$ RMU/CONVERT/CONFIRM MF_PERSONNEL
%RMU-I-RMUTXT_000, Executing RMU for Oracle Rdb V7.3-320 on OpenVMS Alpha V8.4
Are you satisfied with the backup of USER$DISK1[TESTING]MF_PERSONNEL.RDB;1 and
any associated after image (.aij) files [N]? Y
%RMU-I-AIJ_DISABLED, after-image journaling is being disabled temporarily for
the Convert operation
%RMU-I-LOGCONVRT, database root converted to current structure level
%RMU-S-CVTDBSUC, database USER$DISK1:[TESTING]MF_PERSONNEL.RDB;1 successfully
converted from version V7.1 to V7.3
%RMU-I-CVTCOMSUC, CONVERT committed for USER$DISK1:[TESTING]MF_PERSONNEL.RDB;1
to version V7.3
%RMU-I-LOGCREAIJ, created after-image journal file USER$DISK1:[TESTING]MFP.AIJ;2
%RMU-I-LOGMODSTR,    activated after-image journal "RDB$JOURNAL"
%RMU-W-DOFULLBCK, full database backup should be done to ensure future recovery
$
```

If you reply YES and the database has multiple fixed size journals, RMU temporarily disables
after–image journaling, converts the database, and then re–enables after–image journaling.

```
$ RMU/CONVERT SAMPLE
%RMU-I-RMUTXT_000, Executing RMU for Oracle Rdb V7.3-320 on OpenVMS Alpha V8.4
Are you satisfied with the backup of USER$DISK1[TESTING]SAMPLE.RDB;1 and any
associated after image (.aij) files [N]? Y
%RMU-I-AIJ_DISABLED, after-image journaling is being disabled temporarily for
the Convert operation
%RMU-I-LOGCONVRT, database root converted to current structure level
%RMU-S-CVTDBSUC, database USER$DISK1[TESTING]SAMPLE.RDB;1 successfully
converted from version V7.1 to V7.3
%RMU-I-CVTCOMSUC, CONVERT committed for USER$DISK1[TESTING]SAMPLE.RDB;1 to
version V7.3
%RMU-I-LOGMODSTR,    activated after-image journal "SAMPLE_JOURNAL_01"
%RMU-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

Please note, in the case of multiple fixed sized journals, RMU Convert will initialize each after–image
journal during the convert of the database. However, if any data exists in the journals which has not
been backed up, RMU will mark the journal as SUPPRESSED. This can be observed using RMU
Dump Header=JOURNAL or through the RMU Show Statistics command.
This setting gives the database administrator an opportunity to save the after image journal from the
prior version. Although that journal can no longer be applied to the converted database, it might be
valuable for other reasons, for example, to unload using RMU Unload After_Journal.
An RMU command similar to the following can be used to unsuppress the journal and have it
returned to the pool of fixed length journals.

```
$ rmu/set after/alter=(Name=SAMPLE_JOURNAL_01) foo
```

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 1.9.1). 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;
```

3. Backup the converted database immediately. The conversion operation creates a database that is different from the original. The after–image journal (aij) files correspond to the newly converted database. If you need to perform an RMU Restore operation, you will need to RMU Recover the after–image journal files against the backup of the new database. For more information about RMU Convert, see the Oracle RMU Reference Manual.

3.9 Tailoring Your System

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

3.9.1 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$LANGUAGES. Refer to the SQL Reference Manual for more information on the LANGUAGE clause of the SQL SET statement and the SYS$LANGUAGES logical name.

3.9.2 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=RDBVMSV7.3 /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.

3.9.3 Using the RDB$REMOTE73 Account for Remote Access

The Oracle Rdb installation creates the RDB$REMOTE73 account specifically for remote access. This account can be used by any program accessing any remote database. Programs that execute on remote nodes and access Oracle Rdb databases on your node through DECnet or TCP/IP can log in to your system through the RDB$REMOTE73 account.

3.9.3.1 DECnet and the RDBSERVER Object

For DECnet, the Oracle Rdb Release 7.3.3.2 installation procedure defines RDB$REMOTE73 as the default account for the RDBSERVER object. This definition supersedes any previous assignment you may have made for the RDBSERVER object.

The RDB$REMOTE73 account includes a password assigned by the system during the installation procedure. The password provided is used for the RDB$REMOTE73 account and in the DECnet object database on your node. This means that the RDB$REMOTE73 password and the password assigned to the RDBSERVER object will be the same. However, in a cluster environment, the installation procedure assigns the same password to the RDB$REMOTE73 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$REMOTE73 account, as long as the remote node allows RDB$REMOTE73 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$REMOTE73 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$REMOTE73 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$REMOTE73 account has the DISUSER flag set, then accessing the database through the RDB$REMOTE73 account will fail. The DISUSER flag disables the RDB$REMOTE73 account.

The RDB$REMOTE73 account is a restricted account. It does not require a SYS$MANAGER:SYLOGIN.COM procedure. However, if you encounter any errors with the use of the RDB$REMOTE73 account, check that the SYS$SYLOGIN logical name (if defined) points to a working SYLOGIN.COM procedure.

RDB$REMOTE73 does require a login procedure. The login procedure for RDB$REMOTE73 is RDB$REMOTE_LOGIN73.COM; it resides in SYS$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$REMOTE73 account login step, you must edit the RDB$REMOTE_LOGIN73.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.

3.9.3.2 TCP/IP and the RDBSERVER Object

For TCP/IP, the Oracle Rdb installation procedure defines RDB$REMOTE73 as the default account for the TCP/IP RDBSERVER object if the TCP/IP utility is installed at that time. If TCP/IP is not present when Oracle Rdb is installed, you must manually define the RDBSERVER object in TCP/IP. See Section 4.2.2 for an explanation of setting up TCP/IP services for remote access.

3.9.3.3 Network Accounts

Because the Hot Standby functionality requires a network object server (RDMAIJ73) to facilitate communications between the master and the standby database, the Hot Standby software automatically creates an RDMAIJ73 account and object. The installation procedure asks you to supply a valid user identifier for this account.

3.9.3.4 Network Protocols

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

3.9.3.4.1 DECnet

For Hot Standby, 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.

3.9.3.4.2 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 RDMAIJ73 service:

```
$TCPIP SET SERVICE RDMAIJ73
    /PORT=n
    /USER_NAME=RDMAIJ73
    /PROCESS_NAME=RDMAIJ73
    /FILE=SYS$SYSTEM:rdmaijserver73.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:

```
$TCPIP ENABLE SERVICE RDMAIJ73
```

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
```

3.9.3.5 Privileges

For security reasons, the AIJSERVER account (RDMAIJ73) 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.

3.9.4 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 RDMSTT73 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.

3.9.4.1 DECnet

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

3.9.4.2 TCP/IP

Cluster wide statistics support for 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 RDMSTT73 service:

```
$TCPIP SET SERVICE RDMSTT73
    /PORT=n
    /USER_NAME=RDMSTT73
    /PROCESS_NAME=RDMSTT73
    /FILE=SYS$SYSTEM:RDMSTTSERVER.COM
    /LIMIT=m
```

where n is an available port number (use the same port number on all nodes), and m is the number of concurrent connections.

2. Enable the service:

```
$TCPIP ENABLE SERVICE RDMSTT73
```

3. Define RDM$BIND_SST_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.

3.9.5 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]RDB073_73_FILES.DAT
```

3.10 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:RMONSTART73.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 RMONSTART73.COM and SQL$STARTUP.COM located in the SYS$STARTUP directory.

3.11 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
```

3.12 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$IVP73
```

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

3.13 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;
```

3.14 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 7.0 (Multiversion - RDB V7.0-9)
      2        Version 7.1 (Multiversion - RDB V7.1-521)
      3        Version 7.2 (Multiversion - RDB V7.2-570)
      4        Version 7.3 (Multiversion - RDB V7.3-310)
```

```
    0      Quit

  Enter Choice to deinstall (0...4) :
```

Enter the menu number for the version you want to delete. For example, to delete release 7.1 Multiversion, enter the following:

```
Enter Choice to deinstall (0...4) : 2
```

The command procedure displays the following message:

```
You are about to deinstall Rdb 7.1 (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 RMONSTOP71.COM.
If the Rdb Version 7.1 (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 7.1 (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 7.1 (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 7.1 (Multiversion) monitor started on node SYSTEM3
```

Regardless of whether you enter YES or NO, the command procedure creates the RDB$CLUSTER_DEINSTALL71.COM command procedure in your SYS$SCRATCH directory. Use this command procedure to deinstall Oracle Rdb Release 7.1 (Multiversion) from other nodes in the cluster. You must either run this command procedure on each node that has release 7.1 (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 7.1 (Multiversion): YES
```

The final prompt asks you whether or not you want to delete the RDB$REMOTE71 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$REMOTE71? [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 7.1 (Multiversion) now complete
```

For versions of Oracle Rdb after 7.3.3.1 (ie, starting with 7.3.3.2), the deinstallation procedure will also display messages from the PCSI product show history tool showing that the version you have deinstalled will be listed as removed when issuing the Product Show History command.

```
The following product has been selected:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)

Do you want to continue? [YES]

The following product will be removed from destination:
    ORCL I64VMS RDB73 V7.3-320              DISK$CLYPPR84_2:[VMS$COMMON.]


Portion done: 0%
...100%

The following product has been removed:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)
```

3.15 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.

3.16 PCSI Support for Rdb Kit Installation and Deinstallation

Whenever Oracle Rdb is installed or deinstalled, Oracle Rdb will be registered in the PCSI software product database. This will allow users to use the PCSI PRODUCT SHOW HISTORY and PRODUCT SHOW PRODUCT commands to display information about releases of Oracle Rdb that have been installed or deinstalled. This information can be helpful as input whenever a Service Request (SR) is submitted to Oracle Support.

The following lines will be displayed during the installation of Oracle Rdb, showing that the installation has been registered in the PCSI database.

```
The following product has been selected:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)

The following product will be registered:
    ORCL I64VMS RDB73 V7.3-320              DISK$NODE84_2:[VMS$COMMON.]

File lookup pass starting ...

Portion done: 0%
...100%

File lookup pass completed search for all files listed in the product's PDF
Total files searched: 0   Files present: 0   Files absent: 0

The following product has been registered:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...
```

Registration in the PCSI software product database allows a user to use commands such as the following to track what Oracle Rdb releases are currently installed and the history of any past product installations and deinstallations.

```
$ PRODUCT SHOW HISTORY/SINCE
------------------------------------ ----------- ----------- --- -----------
PRODUCT                              KIT TYPE    OPERATION   VAL DATE
------------------------------------ ----------- ----------- --- -----------
ORCL I64VMS RDB73 V7.3-320           Transition  Reg Product (U) 10-OCT-2019
------------------------------------ ----------- ----------- --- -----------

1 item found

$ PRODUCT SHOW HISTORY RDB7*
------------------------------------ ----------- ----------- --- -----------
PRODUCT                              KIT TYPE    OPERATION   VAL DATE
------------------------------------ ----------- ----------- --- -----------
ORCL I64VMS RDB73 V7.3-320           Transition  Reg Product (U) 10-OCT-2019
------------------------------------ ----------- ----------- --- -----------

1 item found

$ PRODUCT SHOW PRODUCT RDB7*
------------------------------------ ----------- ---------
PRODUCT                              KIT TYPE    STATE
------------------------------------ ----------- ---------
```

```
ORCL I64VMS RDB74 V7.3-320            Transition  Installed
----------------------------------- ----------- ---------

1 item found
```

The following lines will be displayed during the deinstallation of Oracle Rdb, showing that the removal of the release has been registered in the PCSI database. Deinstallation is performed by executing the DCL procedure SYS$MANAGER:RDB$DEINSTALL_DELETE.COM. Please refer to Section 3.14 for further details.

```
The following product has been selected:
    ORCL I64VMS RDB73 V7.3-320            Transition (registration)

The following product will be removed from destination:
    ORCL I64VMS RDB73 V7.3-320            DISK$CLYPPR84_2:[VMS$COMMON.]

Portion done: 0%...100%

The following product has been removed:
    ORCL I64VMS RDB74 V7.3-320            Transition (registration)
```

The example below shows the information that will be displayed by the PCSI PRODUCT commands as a result of the deinstallation of a release of Oracle Rdb.

```
$ PRODUCT SHOW HISTORY/SINCE
----------------------------------- ----------- ----------- --- -----------
PRODUCT                             KIT TYPE    OPERATION   VAL DATE
----------------------------------- ----------- ----------- --- -----------
ORCL I64VMS RDB73 V7.3-320          Transition  Remove       -  10-OCT-2019
ORCL I64VMS RDB73 V7.3-320          Transition  Reg Product (U) 10-OCT-2019
----------------------------------- ----------- ----------- --- -----------
2 items found

$ PRODUCT SHOW HISTORY RDB7*
----------------------------------- ----------- ----------- --- -----------
PRODUCT                             KIT TYPE    OPERATION   VAL DATE
----------------------------------- ----------- ----------- --- -----------
ORCL I64VMS RDB73 V7.3-320          Transition  Remove       -  10-OCT-2019
ORCL I64VMS RDB73 V7.3-320          Transition  Reg Product (U) 10-OCT-2019
----------------------------------- ----------- ----------- --- -----------
2 items found

$ PRODUCT SHOW PRODUCT RDB7*
----------------------------------- ----------- ---------
PRODUCT                             KIT TYPE    STATE
----------------------------------- ----------- ---------
0 items found
```

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.

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

4.1 Access Remote Databases from SQL

The SQL Database language provides several commands that reference a database file specification.

These commands include:

- CREATE DATABASE Statement
- ALTER DATABASE Statement
- DROP DATABASE Statement
- ATTACH Statement
- CONNECT Statement
- DECLARE ... ALIAS Statement

These commands provide several ways to reference a remote database and provide the credentials required to access that remote system and database.

- Provide the DECnet credentials in the node specification

  ```
  SQL> ATTACH 'FILENAME lulu"username password"::dev:[dbdir]dbrootfile';
  ```

  The remote credentials may reference the RDB$REMOTE73 username or provide an alternate OpenVMS username. See Section 4.2 for details on setting up RDB$REMOTE73 user and server.
- Provide the credentials to Rdb for either DECnet or TCP/IP access to the database with the USER and USING clauses.

  ```
  SQL> ATTACH 'FILENAME lulu.xyzzy.com"username password"::dev:[dbdir]dbrootfile –
  cont>  USER ''username'' USING ''password'' ';
  ```
- Provide the credentials in the configuration file using the SQL_USERNAME and SQL_PASSWORD parameters. (See Section 4.4.1 for more information on configuration files).
- Do not provide explicit credentials, but instead use proxy access. This requires that the system administrator defines the appropriate definitions in the OpenVMS proxy database using the AUTHORIZE utility. Please refer to the *HPE OpenVMS System Management Utilities Reference Manual* for instructions about the AUTHORIZE utility).

  ```
  SQL> ATTACH 'FILENAME lulu::dev:[dbdir]dbrootfile';
  ```

  Oracle Rdb supports proxy access when the transport is TCPIP and OpenVMS supports proxy access for DECnet. The proxy definition defines which user may be impersonated on the remote system for the incoming client node and user. Further, the proxy definition may list additional users that can be specified by the USER clause that may be impersonated.

  ```
  SQL> ATTACH 'FILENAME lulu::dev:[dbdir]dbrootfile USER ''JJONES'' ';
  ```

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

4.2 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$REMOTE73 server account to allow remote access to Oracle Rdb databases. The RDB$REMOTE73 account can be used by any program accessing any remote database on OpenVMS.

The Oracle Rdb installation attempts to set up a service for HPE TCP/IP Services for OpenVMS. If you are using a TCP/IP product other than HPE TCP/IP Services for OpenVMS, refer to that products 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$REMOTE73 account with the OpenVMS Authorize (AUTHORIZE) utility
- Enable the RDB$REMOTE73 account in the OpenVMS Authorize utility

4.2.1 Setting Up Remote Access in DECnet Phase IV

You must have the RDB$REMOTE73 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 4.2.1.1.
2. The password of the RDB$REMOTE73 account matches the password of the RDBSERVER DECnet object. See Section 4.2.1.2.
3. The RDB$REMOTE73 account exists. Use the OpenVMS Authorize utility (AUTHORIZE). See Section 4.2.3.

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

4.2.1.1 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:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW OBJECT RDBSERVER
Object Volatile Summary as of 23-MAY-2018 12:59:04

  Object    Number  File/PID                   User Id          Password

 RDBSERVER     35  RDBSERVER.COM               RDB$REMOTE73     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:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW OBJECT RDBSERVER CHARACTERISTICS

Object Volatile Characteristics as of 23-MAY-2019 13:01:05

Object = RDBSERVER

Number              = 35
File id             = RDBSERVER.COM
User id             = RDB$REMOTE73
Account             = RDB$REMOTE73
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$SYSTEM:RDBSERVER73.EXE and SYS$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.

4.2.1.2 Verifying Matching Passwords for the RDB$REMOTE73 Account in SYSUAF and for the RDBSERVER DECnet Object in the NCP Utility

The password for the RDB$REMOTE73 account in the user authorization file (SYSUAF) 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 SYSUAF and NCP.

Simply looking at the password for the RDBSERVER DECnet object in the NCP utility and then setting the RDB$REMOTE73 password in SYSUAF 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:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> MODIFY RDB$REMOTE73/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
$
$ RUN SYS$SYSTEM: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$REMOTE73 account and RDBSERVER DECnet object.

4.2.1.3 Setting Up Remote Access in DECnet−Plus

You must have the RDB$REMOTE73 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$REMOTE73 account exists. Use the OpenVMS Authorize (AUTHORIZE) utility. Section 4.2.3 provides more detail about the RDB$REMOTE73 account.
2. The RDB$REMOTE73 account is enabled.
3. The RDBSERVER DECnet object number 35 is present in the NCL utility.
   If the RDBSERVER DECnet object does not exist, you must install Oracle Rdb. Section 4.2.1.4 explains how to verify if the DECnet object is present. Refer to Chapter 2 for installation procedures.
4. The status of proxy access is appropriate.
   To allow remote node access to a database on your system, set the proxy access for the RDBSERVER DECnet object to incoming using the NCL utility. To allow access to a database on a remote node, set the proxy access to outgoing. Allowing access to and from your system is the default. Section 4.2.1.4 and Section 4.2.1.5 explain how to check and change the status of the proxy access.
5. Database privileges exist for RDB$REMOTE73. Section 4.3.1 describes how to grant database privileges for remote access.
6. The proxy accounts are set up to avoid displaying the RDB$REMOTE73 password. The Oracle Rdb7 Guide to SQL Programming describes how to attach to a remote database through a proxy account.
7. The LOGIN.COM procedures for the RDB$REMOTE73 account and any proxy accounts contain the appropriate commands if you want product−specific files to be run during the RDB$REMOTE73 login step. Section 3.9.3 and Section 2.4.7 provide information on RDB$REMOTE_LOGIN73.COM and LOGIN.COM procedures for proxy accounts.

4.2.1.4 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:

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

Node 0 Session Control Application RDBSERVER
at 2014-01-09-16:33:28.790-04:00

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$SYSTEM:RDBSERVER.COM
    User Name                       = "RDB$REMOTE73"
    Incoming OSI TSEL               = <Default value>
NCL> EXIT
```

4.2.1.5 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 2014-01-09-08:50:16.490-04:00

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 2014-01-09-08:50:36.320-04:00

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.

4.2.2 Setting Up Remote Access in TCP/IP Services

The TCP/IP network protocols can be used to access remote Oracle Rdb V6.1 and later databases. To do this, you must have the TCP/IP service RDBSERVER defined with the TCPIP utility. The Oracle Rdb installation procedure will automatically set up and enable the RDBSERVER service if the TCPIP 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 the following:

1. The existence of the RDB$REMOTE73 account using the OpenVMS Authorize (AUTHORIZE) utility. Section 4.2.3 provides more detail about the RDB$REMOTE73 account.
2. The presence of the RDBSERVER service in the TCPIP utility.

4.2.2.1 Verify the Presence of the RDBSERVER Service

To verify the presence of the RDBSERVER service, you use the TCPIP 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$REMOTE73, and file SYS$SYSTEM:RDBSERVER_TCPIP.COM.

```
$ TCPIP
TCPIP> show service rdbserver/full

Service: RDBSERVER
                        State:      Enabled
Port:            611    Protocol:   TCP              Address:   0.0.0.0
Inactivity:        5    User_name: RDB$REMOTE73      Process:   RDB73
Limit:            10    Active:      0               Peak:          1

File:           SYS$COMMON:[SYSEXE]RDBSERVER_TCPIP.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
```

4.2.2.2 Set Up the RDBSERVER Service

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

```
$ TCPIP
TCPIP> set service RDBSERVER -
    /port=611 -
    /user_name=RDB$REMOTE73 -
    /process=RDB73 -
    /limit=10 -
    /file=SYS$COMMON:[SYSEXE]RDBSERVER_TCPIP.COM
TCPIP> enable service RDBSERVER
TCPIP> exit
```

The value for /LIMIT must be greater than the expected number of simultaneous connects. For more information, see Section 4.2.2.3.

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

```
TCPIP> enable service RDBSERVER
```

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

Check the OpenVMS file protections on the SYS$SYSTEM:RDBSERVERnn.EXE (where nn would be an Oracle Rdb release number) and SYS$SYSTEM:RDBSERVER_TCPIP.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.

4.2.2.3 Changing the TCPIP /LIMIT Defaults

On a given OpenVMS node running TCP/IP, the /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 using TCPIP, log into a privileged account and issue the following commands:

```
$ TCPIP
TCPIP> disable service RDBSERVER
TCPIP> set service RDBSERVER /LIMIT=64
TCPIP> enable service RDBSERVER
TCPIP> exit
```

4.2.2.4 Domain–qualified TCP/IP Node Names in Distributed Transactions

When using TCP/IP for Oracle Rdb remote connections, distributed transactions involving databases on nodes which are not on the same subnet may not work.

Remote Rdb has the capability to make remote connections via TCP/IP in lieu of DECnet. However, distributed transactions involving remote databases connected to via TCP/IP have been difficult. This is because Rdb relies on OpenVMS DECdtm for distributed transaction support and DECdtm requires DECnet for off–node communication. (This is an OpenVMS and not an Rdb restriction. Contact Hewlett–Packard OpenVMS Support for more details.)

OpenVMS provides a capability to run DECnet over TCP/IP so that OpenVMS services which require DECnet (like DECdtm) can operate in an environment where a TCP/IP network is used as the communications backbone. This capability allows DECdtm (and hence Rdb) to manage distributed transactions via TCP/IP. (See HP's OpenVMS DECnet–Plus documentation set for how to configure and use this capability.)

However, for a transaction involving a remote database, Rdb only provides the SCSNODE name of the remote node to DECdtm. For example, consider the following SQL attaches to two remote databases using TCP/IP:

```
SQL> attach 'alias db1 filename node1.a.b.c::db_root:db1 user ''me'' using
''pw''';
SQL> attach 'alias db2 filename node1.a.b.c::db_root:db2 user ''me'' using
''pw''';
```

In the above example, Rdb can successfully connect to both remote databases using the TCP/IP address "node1.a.b.c." but when multiple databases are attached, Rdb implicitly uses distributed transactions via DECdtm. Since Rdb only passes DECdtm the SCSNODE name retrieved from the RDBSERVERnn at the other end of the connection, DECdtm does not, in general, have the information it needs to resolve the remote reference. It will only be able to do so if the SCSNODE name and the TCP/IP node name are the same and the local node is on the same subnet (i.e. ".a.b.c" in the example). Otherwise, after the second attach is made, the following error message will be received as soon as a transaction is started:

```
SQL> set trans read write;
%RDB-F-SYS_REQUEST_CAL, error from system services request - called from 100001
-RDB-E-DECDTMERR, DECdtm system service call error
-IPC-E-BCKTRNSFAIL, failure on the back translate address request
```

WORKAROUND

There are three potential workarounds:

- If distributed transactions are unimportant to the application, they can be disabled by defining the logical name SQL$DISABLE_CONTEXT to TRUE. Rdb will then not call DECdtm and the node name resolution problem will not be seen. However, it will be the problem of the application to maintain database integrity in the event that a commit succeeds on one database and not on another. See the Rdb Guide to Distributed Transactions for more information.
- If all the nodes involved in the distributed transaction are in the same domain, then TCP/IP can resolve the node with only the first part of the node provided that the SCSNODE name is identical to it. In the example above, this would mean that the remote node had an SCSNODE name of "NODE1" and that the local node was on TCP/IP subnet ".a.b.c".

- It may also be possible to define a DNS/BIND alias name for the remote node's SCSNODE name to the local node's TCP/IP database. This should allow the SCSNODE name passed by Rdb Dispatch to be translated successfully. For example, assuming HP TCP/IP Services for OpenVMS is the TCP/IP protocol stack then a command like the following could be used on the local node:

```
$ TCP SET HOST NODE1.A.B.C/address=nnn.nnn.nnn.nnn/alias=NODE1_SCS
```

Where "nnn.nnn.nnn.nnn" is the IP address and "NODE1_SC" the OpenVMS SCSNODE name of the remote node. See the HP DECnet–Plus documentation set for more information on how to maintain TCP/IP domain databases.

4.2.3 Verifying the Setup of the RDB$REMOTEnn Account with the OpenVMS Authorize Utility

Use the OpenVMS Authorize (AUTHORIZE) utility to determine if the RDB$REMOTE73 account exists on your system. You must have the system user identification code (UIC) or the SYSPRV privilege to run AUTHORIZE. For example:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> SHOW RDB$REMOTE73

Username: RDB$REMOTE73                          Owner:
Account:                                        UIC:    [250,201] ([RDB$REMOTE73])
CLI:      DCL                                   Tables: DCLTABLES
Default:  SYS$COMMON:[RDB$REMOTE73]

LGICMD:   SYS$SYSTEM:RDB$REMOTE_LOGIN73.COM
Flags:  Disctly DefCLI Lockpwd Dismail Disreconnect
   .
   .
   .
```

If the RDB$REMOTE73 account does not exist, you must install Oracle Rdb. Refer to Chapter 2 for installation procedures.

4.3 Granting Database Privileges for Remote and Network Access

This section describes how to grant database privileges to the RDB$REMOTE73 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, view, sequence, module, procedure or function 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, or object. 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, view, sequence, module, procedure or function.

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

```
$ SQL$
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,JJONES],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, or other database objects.

However, when Oracle Rdb creates a database it does not automatically give the RDB$REMOTE73 account any access rights to it. Thus, to enable a database for remote access, you must grant it database privileges with the GRANT statement. See Section 4.3.1 for information about granting database privileges to the RDB$REMOTE73 account to allow remote access to a database.

See Section 4.3.2 for information about controlling privileges for the NETWORK identifier for network access.

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

4.3.1 Granting Database Privileges to the RDB$REMOTE73 Account for Remote Access

Oracle Rdb does not give the RDB$REMOTE73 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$REMOTE73 account the SELECT privilege only, type the following:

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

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

SQL> GRANT SELECT ON DATABASE RDB$DBHANDLE TO RDB$REMOTE73;
SQL> SHOW PROTECTION ON DATABASE RDB$DBHANDLE;

Protection on Alias RDB$DBHANDLE
    (IDENTIFIER=[RDB$REMOTE73],ACCESS=SELECT)
    (IDENTIFIER=[SQL,JJONES],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$REMOTE73 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.

4.3.2 Controlling Database Privileges for Network Access

When a client application accesses a remote database, it is running as an impersonated user on the remote system. Therefore, access control lists on the database should be defined to control access for the impersonated user, which may not necessarily be the client user running the application.

OpenVMS implicitly creates special rights identifiers for each created process depending on how it is created: INTERACTIVE, BATCH, REMOTE and NETWORK. The remote server process will be granted NETWORK, so if the database is defined as PROTECTION IS ACL, this special rights identifier may be used to control access through the remote connection.

```
$ SHOW PROC/RIGHTS

16-SEP-2018 09:40:24.70   User: JJONES              Process ID:    2143A6E4
                          Node: LULU                Process name: PROJECT_10

Process rights:
 JJONES                                  resource
 INTERACTIVE
 REMOTE
 DEV_MANAGER

System rights:
 SYS$NODE_LULU
$
```

  • Creating a general restriction for NETWORK
    In this example, the system administrator wishes to limit remote access to SELECT privilege on the
    database. This will allow applications to attach to the database but will prevent a remote session from

creating or altering database objects.

```
SQL> GRANT SELECT ON DATABASE PERS TO NETWORK;
SQL> SHOW PROTECTION ON DATABASE PERS;
Protection on Alias PERS
    (IDENTIFIER=NETWORK,ACCESS=SELECT)
    (IDENTIFIER=[SQL,JJONES],ACCESS=SELECT+INSERT+UPDATE+DELETE+
      SHOW+CREATE+ALTER+DROP+DBCTRL+OPERATOR+DBADM+REFERENCES)
    (IDENTIFIER=[*,*],ACCESS=SELECT)
SQL> COMMIT;
SQL> DISCONNECT DEFAULT;
SQL> EXIT
```
- Modifying access for a specific user
This example adds the NETWORK right to the user. When that user accesses the database remotely,
they will have some limited access to a database table.

```
SQL> GRANT SELECT,INSERT,UPDATE,DELETE
cont>  ON TABLE EMPLOYEES TO [JJONES]+NETWORK;
SQL> COMMIT;
```

Alternately, you can specifically prevent access to a table or other object. First, create the entry by
granting some privilege (any initial privilege would be sufficient), and then revoke all privileges.
These two steps are required to create an access control entry for that user and rights combination.

```
SQL> GRANT SHOW ON TABLE EMPLOYEES TO [JJONES]+NETWORK;
SQL> REVOKE ALL PRIVILEGES ON TABLE EMPLOYEES FROM [JJONES]+NETWORK;
SQL> SHOW PROTECTION ON TABLE EMPLOYEES;
Protection on Table EMPLOYEES
    (IDENTIFIER=[TEAM,J_JONES]+NETWORK,ACCESS=NONE)
    (IDENTIFIER=[APPS,MANAGER],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;
```

4.3.3 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 ATTACH statement. The server account on the remote node must be able to
access the database first, before being able to impersonate the specified user. This remains true for the
duration of your remote session.

The server account may be RDB$REMOTE73, a username specific to an application service (see Section
4.2.2.2), or a DECnet proxy for the current user.

4.4 Improving Performance When Attaching to a Remote Database

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

4.4.1 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.

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$CLIENT_DEFAULTS.DAT	SQL_ALTERNATE_SERVICE_NAME SQL_DEFAULTS_RESTRICTION SQL_ENABLE_PROBE SQL_ENABLE_TCPIP_PROXY SQL_MESSAGE_VECTOR_RETURN_TYPE SQL_NETWORK_BUFFER_SIZE SQL_NETWORK_NUMBER_ATTACHES SQL_NETWORK_TRANSPORT_TYPE SQL_PASSWORD SQL_RCV_PREFETCH_ROWS SQL_RETRY_OLD_PROTOCOL SQL_SGS_PREFETCH_ROWS SQL_TRANS_START_WAIT SQL_USERNAME
Server	RDB$SERVER_DEFAULTS.DAT	SQL_ALTERNATE_SERVICE_NAME SQL_DEFAULTS_RESTRICTION SQL_ENABLE_TCPIP_PROXY SQL_NETWORK_BUFFER_SIZE

The SQL_ALTERNATE_SERVICE_NAME, SQL_ENABLE_TCPIP_PROXY,

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	Client or server
SQL_ENABLE_PROBE	TRUE FALSE	FALSE	Client only
SQL_ENABLE_TCPIP_PROXY	TRUE FALSE	TRUE	Client or server
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_RETRY_OLD_PROTOCOL	TRUE FALSE	FALSE	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 4.4.2) 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$SYSTEM_DEFAULTS logical name. If it finds a configuration file, it reads the file to check the values assigned to the parameters that are specified. It checks the SQL_DEFAULTS_RESTRICTION parameter because that parameter determines whether Oracle Rdb also reads any other configuration files located in the directories defined by the RDB$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
SQL_ENABLE_TCPIP_PROXY FALSE
```

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_ENABLE_TCPIP_PROXY	TRUE	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_RETRY_OLD_PROTOCOL	FALSE	FALSE
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, the SQL_RCV_PREFETCH_ROWS parameter value from 20 to 50, and disables TCPIP proxy access. 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.

4.4.2 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). Keywords and parameter values may be specified in either lower or uppercase. String values may be quoted. Comments can be added to the configuration file preceded by a "!" character.

For example, the following RDB$CLIENT_DEFAULTS.DAT client configuration file changes the defaults for three parameters:

```
! Current defaults 24 November, 2019
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 omit a parameter
- 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.*

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

4.4.2.1 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 TCP/IP 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 TCP/IP service.

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

Note

> *When creating alternate services, ensure that the service name and port number are the same on all nodes that specify the service. The configuration file references the name but TCP/IP uses the port for communication across the network.*

4.4.2.2 Specifying SQL_DEFAULTS_RESTRICTION

The SQL_DEFAULTS_RESTRICTION parameter controls the processing of default characteristics for the system, group, or user. You can use the SQL_DEFAULTS_RESTRICTION parameter in a client or server configuration file.

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

Oracle Rdb uses the values specified to limit the configuration files processed during startup. For example, after loading SQL_DEFAULTS_RESTRICTION from the system configuration file, if it specifies SYSTEM then no further configuration files will be opened. On the other hand, if the value is GROUP (and this is not overridden) then once the group configuration file is processed no further configuration files will be opened.

Refer to Section 4.4.1 for detailed information about how Oracle Rdb uses the SQL_DEFAULTS_RESTRICTION parameter.

4.4.2.3 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 access 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.

4.4.2.4 Specifying SQL_ENABLE_TCPIP_PROXY

The SQL_ENABLE_TCPIP_PROXY parameter turns on and off TCP/IP proxy access support. By default, proxy access through TCP/IP is enabled but can be disabled by setting SQL_ENABLE_TCPIP_PROXY to FALSE in one of the configuration files. The parameter can be present in either the client or the server configuration files.

Table 4–2 provides key information about the SQL_ENABLE_TCPIP_PROXY parameter.

4.4.2.5 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 server is not installed on the client system. To overcome this condition, you can set the SQL_MESSAGE_VECTOR_RETURN_TYPE parameter to TEXT.

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

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.

4.4.2.6 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 client and server configuration 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.

4.4.2.7 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.

4.4.2.8 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.

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

! 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.

4.4.2.9 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.

4.4.2.10 Specifying SQL_RETRY_OLD_PROTOCOL

The SQL_RETRY_OLD_PROTOCOL parameter turns on and off the "retry using old communication protocol". By default, the retry is disabled but can be enabled by setting SQL_RETRY_OLD_PROTOCOL to TRUE in the client configuration file.

Table 4–2 provides key information about the SQL_RETRY_OLD_PROTOCOL parameter.

4.4.2.11 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.

4.4.2.12 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.

4.4.2.13 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.

4.4.3 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
```

4.4.4 Controlling the Timing Out of RDBSERVER Processes

Bug 4278533

When using DECnet to connect to a remote database, OpenVMS will typically use NETSERVER.COM or NET$SERVER.COM to start the remote RDBSERVER process. A corresponding log file should be created (for example NETSERVER.LOG or NET$SERVER.LOG).

The RDBSERVER processes are reusable. A timeout clock starts when the link is disconnected. If another connect, using the same access control, is received before the server times out then the server will be reused. Otherwise, the server exits and another one is started when the next connect request is received.

The timeout period (the time from when the last disconnect is done to when the server actually exits) can be controlled by defining the system logical NETSERVER$TIMEOUT. The logical uses a delta time value "dddd hh:mm:ss.ss".

For example, to set the timeout to 9 minutes use:

```
$ define/system  NETSERVER$TIMEOUT  "0 00:09:00.00"
```

If NETSERVER$TIMEOUT is not defined, then the default value of 5 minutes is used. If NETSERVER$TIMEOUT is set to "0 00:00:00.00" then there will be no timeout. The service will exit on disconnect and will not be reused.

This reuse feature does not appear to be available when using TCP/IP.

4.4.2.12 Specifying SQL_USERNAME and SQL_PASSWORD   101

4.5 Troubleshooting for Remote Access

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

4.5.1 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.

4.5.2 Remote Attach Stalls Before Detecting a Node is Unreachable

A remote attach can stall for a noticeable period, typically 75 seconds, before detecting a node is unreachable.

The following example shows the expected error message when attempting to access a database on a node that is not reachable. The problem is that when the value of the parameter SQL_NETWORK_TRANSPORT_TYPE in the file RDB$CLIENT_DEFAULTS.DAT is not specifically set to DECNET (or not specified at all), a stall of typically 75 seconds will happen before you get an error message.

```
SQL> attach 'file 1::disk1:[dbdir]db';
%SQL-F-ERRATTDEC, Error attaching to database 1::disk1:[dbdir]db
-RDB-F-IO_ERROR, input or output error
-SYSTEM-F-UNREACHABLE, remote node is not currently reachable
```

When connecting via DECnet, the error is typically returned instantly so a significant stall will not be seen in this case. Failing to define SQL_NETWORK_TRANSPORT_TYPE will result in connecting via the DEFAULT method which is to first try connecting via DECnet and if that fails attempt to connect via TCPIP and hence a 75 second stall will take place unless TPC_KEEPINIT is set to a value lower than 150.

There are two possible ways to avoid the stall and get the error message after a user configurable period of time or instantly: decrease the value of the TCPIP parameter TCP_KEEPINIT or explicitly specify SQL_NETWORK_TRANSPORT_TYPE as DECNET.

The default behavior when attempting to connect to an unreachable node via TCPIP is to stall 75 seconds before returning an error. The stall time is configurable in TCPIP via the parameter TCP_KEEPINIT which is expressed in units of 500 ms. The default value of TCP_KEEPINIT is 150 which corresponds to a 75 second

stall.

4.5.3 Engine Error Logging

This feature allows error messages returned from a database engine on a remote server to be logged. Only non success messages are logged. The server must be running Rdb Release 7.3.2.1 or higher.

These messages will typically be written into a NETSERVER.LOG file. However, they can be written to a different log file by creating an RDB$SERVER_DEFAULTS.DAT file on the server and defining:

```
RCI_DUMP_LOGFILE   "DISK:[DIR]FILE.LOG"
```

This feature is "OFF" by default. It can be turned on by the following methods.

Note: setting any of these methods to "ON" will turn the feature on. Each method can only be set to "TRUE" or "ON". All other values are ignored. Thus, if any one is set "ON" then the feature will be enabled even if another is set "OFF".

1. Define the logical RDB$RDBSHR_ENGINEERR_LOG "ON". The logical must be set on the server so that it is visible to Dispatch. Setting it in the system table may be best. See the following example.

   ```
   DEFINE/SYSTEM   RDB$RDBSHR_ENGINEERR_LOG   "ON"
   ```

2. Create an RDB$SERVER_DEFAULTS.DAT file on the server and define the logical.

   ```
   RCI_ENGINEERR_LOG   "ON"
   ```

3. Create an RDB$CLIENT_DEFAULTS.DAT file on the client and define the logical.

   ```
   RCI_ENGINEERR_LOG   "ON"
   ```

   This will cause the client to instruct the server to turn on Engine Error Logging. This also requires that both client and server are running Rdb Release 7.3.2.1 or higher.

Be aware that SQL_DEFAULTS_RESTRICTION may stop RCI_ENGINEERR_LOG from being read. Thus, it is advised that RCI_ENGINEERR_LOG be in the most privileged .DAT file.

The files are read in the following order:

```
RDB$SYSTEM_DEFAULTS:RDB$SERVER_DEFAULTS.DAT
RDB$GROUP_DEFAULTS:RDB$SERVER_DEFAULTS.DAT
RDB$USER_DEFAULTS:RDB$SERVER_DEFAULTS.DAT
Then SYS$LOGIN:RDB$SERVER_DEFAULTS.DAT is read only if
RDB$USER_DEFAULTS:RDB$SERVER_DEFAULTS.DAT does not exist.

RDB$SYSTEM_DEFAULTS:RDB$CLIENT_DEFAULTS.DAT
RDB$GROUP_DEFAULTS:RDB$CLIENT_DEFAULTS.DAT
RDB$USER_DEFAULTS:RDB$CLIENT_DEFAULTS.DAT
Then SYS$LOGIN:RDB$CLIENT_DEFAULTS.DAT is read only if
RDB$USER_DEFAULTS:RDB$CLIENT_DEFAULTS.DAT does not exist.
```

If Engine Error Logging is enabled, an entry is written to the "Keyword values negotiated between client and server..." section of the log file.

This entry indicates Engine Error Logging is on:

```
LOGGING ENGINE ERRORS
```

An example of an error report:

```
** 17-MAY-2018 01:37:24.73: %RDB-E-ENGINEERR, The database engine has returned
an error for client 15a250 connection 21
%RDB-F-SYS_REQUEST, error from system services request
%RDMS-F-FILACCERR, error opening storage area file DISK1:[DATABASE]JOBS.SNP;1
***** Error while processing RCI_CLASS_REQ
```

4.5.4 Troubleshooting Application–Specific Problems

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

4.5.4.1 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 they 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.

4.5.4.2 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 SQL Reference Manual.

4.5.5 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$REMOTE73 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 transaction.	Grant the DBADM and DISTRIBTRAN privileges to the RDB$REMOTE73 account on all databases. See the OpenVMS documentation for more information.
DECdtm is not installed on your system	DECdtm is not started on one or both of the nodes.	Start DECdtm on both nodes if the platform on both nodes is OpenVMS.
	The user application is trying to attach to more than one database, and SQL attempts to start a distributed transaction	Define the logical name SQL$DISABLE_CONTEXT to be TRUE. See the Oracle Rdb Guide to Distributed Transactions.

	by default.	
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 remote node	User application attempted to access a remote database while the RDB$REMOTE73 account was disabled.	Disable the DISUSER flag in the RDB$REMOTE73 account.
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.

4.6 Domain−Qualified TCP/IP Node Names in Distributed Transactions

When using TCP/IP for Oracle Rdb remote connections, distributed transactions involving databases on nodes which are not on the same subnet may not work.

Remote Rdb has the capability to make remote connections via TCP/IP in lieu of DECnet. However, distributed transactions involving remote databases connected to via TCP/IP have been difficult. This is because Rdb relies on OpenVMS DECdtm for distributed transaction support and DECdtm requires DECnet for off−node communication. (This is an OpenVMS and not an Rdb restriction. Contact Hewlett−Packard OpenVMS Support for more details.)

OpenVMS provides a capability to run DECnet over TCP/IP so that OpenVMS services which require DECnet (like DECdtm) can operate in an environment where a TCP/IP network is used as the communications backbone. This capability allows DECdtm (and hence Rdb) to manage distributed transactions via TCP/IP. (See HPE's OpenVMS DECnet−Plus documentation set for how to configure and use this capability.)

However, for a transaction involving a remote database, Rdb only provides the SCSNODE name of the remote node to DECdtm. For example, consider the following SQL attaches to two remote databases using TCP/IP:

```
SQL> attach 'alias db1 filename node1.a.b.c::db_root:db1 user ''me'' using
''pw''';
SQL> attach 'alias db2 filename node1.a.b.c::db_root:db2 user ''me'' using
''pw''';
```

In the above example, Rdb can successfully connect to both remote databases using the TCP/IP address "node1.a.b.c" but when multiple databases are attached, Rdb implicitly uses distributed transactions via DECdtm. Since Rdb only passes DECdtm the SCSNODE name retrieved from the RDBSERVERnn at the other end of the connection, DECdtm does not, in general, have the information it needs to resolve the remote reference. It will only be able to do so if the SCSNODE name and the TCP/IP node name are the same and the local node is on the same subnet (i.e. ".a.b.c" in the example). Otherwise, after the second attach is made, the following error message will be received as soon as a transaction is started:

```
SQL> set trans read write;
%RDB-F-SYS_REQUEST_CAL, error from system services request - called from 100001
-RDB-E-DECDTMERR, DECdtm system service call error
-IPC-E-BCKTRNSFAIL, failure on the back translate address request
```

There are three potential workarounds:

- If distributed transactions are unimportant to the application, they can be disabled by defining the logical name SQL$DISABLE_CONTEXT to TRUE. Rdb will then not call DECdtm and the node name resolution problem will not be seen. However, it will be the problem of the application to maintain database integrity in the event that a commit succeeds on one database and not on another. See the Rdb Guide to Distributed Transactions for more information.
- If all the nodes involved in the distributed transaction are in the same domain, then TCP/IP can resolve the node with only the first part of the node provided that the SCSNODE name is identical to it. In the example above, this would mean that the remote node had an SCSNODE name of "NODE1" and that the local node was on TCP/IP subnet ".a.b.c".

- It may also be possible to define a DNS/BIND alias name for the remote node's SCSNODE name to the local node's TCP/IP database. This should allow the SCSNODE name passed by Rdb Dispatch to be translated successfully. For example, assuming HPE TCP/IP Services for OpenVMS is the TCP/IP protocol stack then a command like the following could be used on the local node:

```
$ TCP SET HOST NODE1.A.B.C/address=nnn.nnn.nnn.nnn/alias=NODE1_SCS
```

Where "nnn.nnn.nnn.nnn" is the IP address and "NODE1_SC" the OpenVMS SCSNODE name of the remote node. See the HPE DECnet−Plus documentation set for more information on how to maintain TCP/IP domain databases.

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 7.3 and was done on a system that had a prior version of Oracle Rdb already installed.

A.1 Sample Installation of Oracle Rdb

```
$ @SYS$UPDATE:VMSINSTAL RDBV73320IM073 DEV:[DIR]
     .
     .
     .
The following products will be processed:

  RDBV73320IM V7.3

        Beginning installation of RDBV73320IM V7.3 at 8:10

No signature manifests found for RDBV73320IM073

* Do you want to install this product [NO]? yes

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
  Copyright © 1995, 2019, Oracle Corporation.  All Rights Reserved.


        Installation procedure for: "Oracle Rdb V7.3-320"

  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$REMOTE73
  account.  The installation procedure will not proceed until you
  enter a valid user identification code (UIC) for the RDB$REMOTE73
  account.  The UIC must be unique.  Format [ggg,mmm].

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

* Enter UIC to be used for RDB$REMOTE73 account: [250,201]

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

  This installation requires the creation of the RDMAIJ73
  account.  The installation procedure will not proceed until you
  enter a valid user identification code (UIC) for the RDMAIJ73
  account.  The UIC must be unique.  Format [ggg,mmm].

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

* Enter UIC to be used for RDMAIJ73 account: [250,202]

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

  This installation requires the creation of the RDMSTT73
  account.  The installation procedure will not proceed until you
  enter a valid user identification code (UIC) for the RDMSTT73
  account.  The UIC must be unique.  Format [ggg,mmm].
```

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

* Enter UIC to be used for RDMSTT73 account: [250,203]

* 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 ...

    OpenVMS IA64 V8.4-2L1 - 21-OCT-2019 08:12

%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$ROOT:[SYSHLP.EXAMPLES.RDB73].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
 VMI$ROOT:[SYSTEST.RDB73].
%VMSINSTAL-I-ACCOUNT, This installation creates an ACCOUNT named RDMSTT73.
%UAF-I-ADDMSG, user record successfully added
%UAF-I-RDBADDMSGU, identifier RDMSTT73 value [000250,000203] added to rights
 database
%VMSINSTAL-I-ACCOUNT, This installation creates an ACCOUNT named RDMSTT73.
%UAF-I-ADDMSG, user record successfully added
%UAF-I-RDBADDMSGU, identifier RDMSTT73 value [000250,000203] added to rights
database
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named RDMSTT73.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
VMI$ROOT:[RDMSTT73].

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

        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:RMONSTART73

        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:RMONSTOP73

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

    %VMSINSTAL-I-ACCOUNT, This installation creates an ACCOUNT named
    RDB$REMOTE73.
    %UAF-I-ADDMSG, user record successfully added
    %UAF-I-RDBADDMSGU, identifier RDB$REMOTE73 value [000250,000201] added
    to rights database
    %VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named
    RDB$REMOTE73.
    %UAF-I-MDFYMSG, user record(s) updated

%VMSINSTAL-I-SYSDIR, This product creates system disk directory
VMI$ROOT:[RDB$REMOTE73].
```

A.1 Sample Installation of Oracle Rdb

```
%VMSINSTAL-I-ACCOUNT, This installation creates an ACCOUNT named RDMAIJ73.
%UAF-I-ADDMSG, user record successfully added
%UAF-I-RDBADDMSGU, identifier RDMAIJ73 value [000250,000202] added to rights
database
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named RDMAIJ73.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
VMI$ROOT:[RDMAIJ73].

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

    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:

      $ @SYS$COMMON:[SYSTEST]RDB$IVP73

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

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

    The release notes for Oracle Rdb are available in the file
    SYS$HELP:RDB07332.RELEASE_NOTES

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


The following product has been selected:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)

The following product will be registered:
    ORCL I64VMS RDB73 V7.3-320              DISK$CLYPPR84_2:[VMS$COMMON.]

File lookup pass starting ...


Portion done: 0%
...100%

File lookup pass completed search for all files listed in the product's PDF
Total files searched: 0   Files present: 0   Files absent: 0

The following product has been registered:
    ORCL I64VMS RDB73 V7.3-320              Transition (registration)

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

 Current PROCESS Oracle Rdb environment is version V7.3-320 (MULTIVERSION)
 Current PROCESS SQL environment is version V7.3-320 (MULTIVERSION)
 Current PROCESS Rdb/Dispatch environment is version V7.3-320 (MULTIVERSION)
 Oracle Rdb monitor (RDMS_MONITOR73) started

 Executing IVP for: Oracle Rdb V7.3-320
```

A.1 Sample Installation of Oracle Rdb   113

```
Current PROCESS Oracle Rdb environment is version V7.3-320 (MULTIVERSION)
Current PROCESS SQL environment is version V7.3-320 (MULTIVERSION)
Current PROCESS Rdb/Dispatch environment is version V7.3-320 (MULTIVERSION)
    Copyright © 1995, 2019, 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

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

    Oracle Rdb V7.3-320

    IVP COMPLETED SUCCESSFULLY

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

A.1 Sample Installation of Oracle Rdb   114

```
IVP completed for Oracle Rdb V7.3-320

        Installation of RDBV73320IM073 V7.3 completed at 8:15

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

    Creating installation data file: VMI$ROOT:[SYSUPD]RDBV73320IM073.VMI_DATA

Enter the products to be processed from the next distribution volume set.
* Products:
        VMSINSTAL procedure done at 08:15
```

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 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.