

Oracle® Rdb

Oracle SQL/Services Release 7.1.5 Release Notes

Release 7.1.0 for OpenVMS Alpha

June 2001

This document contains release note information specific to Oracle SQL/Services, release 7.1.5 for the OpenVMS Alpha operating system shipping with Oracle Rdb release 7.1.0. Also included in this document are release notes pertaining to SQL*Net for Rdb release 7.1.5.

ORACLE

Oracle Rdb Oracle SQL/Services Release 7.1.5 Release Notes, Release 7.1.0 for OpenVMS Alpha

Copyright © 2001, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Rdb, Oracle SQL/Services, Oracle7, Oracle8, PL/SQL, SQL*Net, SQL*Plus, Secure Network Services and, Oracle Rdb7 are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xi
Preface.....	xii
Intended Audience	xii
Operating System Information	xii
Structure.....	xiii
Related Manuals	xiii
Conventions.....	xiii
1 Oracle SQL/Services: New and Changed Features	
1.1 Installing Oracle SQL/Services Release 7.1.5.....	1-1
1.1.1 Save-Set Names	1-1
1.1.2 Software Installation Requirement and Compatibility Information	1-2
1.2 Documentation	1-4
1.3 Summary of Oracle SQL/Services New and Changed Features for Release 7.1.5.....	1-4
1.4 Deprecated Features	1-6
1.5 Obsolete Features	1-10
2 Oracle SQL/Services: Software Errors Fixed	
2.1 Oracle SQL/Services Errors Fixed Since Release 7.0	2-1
2.1.1 Existing Configuration Files Are Preserved During an Installation.....	2-1
2.1.2 Oracle SQL/Services Management Utility Caused a Bugcheck to Occur	2-2

2.1.3	Oracle SQL/Services Dispatcher Failed to Completely Shut Down When There Were Clients Connected.....	2-2
2.1.4	Problems Running Oracle SQL/Services V7.0 Multiversion with a Higher Version of Oracle SQL/Services Standard	2-2
2.1.5	Oracle SQL/Services Executor Incorrectly Returned Multi-Octet Character Set Data from Columns of Type NATIONAL CHARACTER	2-2
2.1.6	Multiversion Installations Erroneously Specified a Version Number.....	2-3
2.1.7	Executors Failed During Startup with RMS-E-PRV Errors.....	2-3
2.1.8	Problem Fetching More Than 65535 Rows in fetch_many Mode.....	2-3
2.1.9	Monitor Process Termination Left Dangling Oracle SQL/Services Processes.....	2-4
2.1.10	All Processes Did Not Explicitly Flush Log File Output	2-4
2.1.11	Installation Did Not Allow DECnet Object Name to Begin with a Numeric Value.....	2-4
2.1.12	Oracle SQL/Services Protocol Message to Monitor Caused a Bugcheck to Occur.....	2-5
2.1.13	Failure While Holding a Mutex During Executor Cleanup	2-5

3 Oracle SQL/Services: Known Problems

3.1	Oracle SQL/Services Manager GUI Known Problems and Restrictions	3-1
3.2	Oracle SQL/Services Release 7.1.5 Server Known Problems and Restrictions.....	3-1
3.2.1	Impersonation Is Not Supported	3-1
3.2.2	Concealed Attributes are Required for Rooted Directory Logicals	3-2
3.2.3	Problems With Installing Oracle Rdb V6.1 After Installing Oracle SQL/Services V7.0 or Higher	3-3
3.2.4	Database Authorization for Database Services.....	3-5
3.2.5	Installation Clarification: How to Determine Ports to Which to Direct Clients....	3-5
3.2.6	How to Recreate Your RMU_DISP Dispatcher and RMU_SERVICE Service.....	3-7
3.2.7	Do Not Kill Oracle SQL/Services Processes	3-9
3.2.8	Oracle SQL/Services Does Not Support Columns That Have a Negative Scale Factor	3-10
3.2.9	Executor Bugchecks if a Pre-V7.0 Oracle SQL/Services Client API Sends an Inappropriate Batch-Execute Abort Message	3-10
3.2.10	Matching UIC Required to Use Oracle Rdb V6.1 With a Database Service That Has Database Authorization Set to Connect User Name.....	3-11
3.2.11	Do Not Shut Down or Restart the SQLSRV_MANAGE System Management Service.....	3-11

3.2.12	Do Not Abbreviate AUTOCONNECT or AUTOSTART Keywords	3-11
3.2.13	Oracle SQL/Services Management Utilities Do Not Prevent Multiple Dispatchers With the Same Port IDs	3-12
3.2.14	Database Service Attached to a Remote Database Does Not Know if the Database is Closed	3-12
3.2.15	Log and Bugcheck Dump File Path Server Attributes are Ignored on OpenVMS Systems.....	3-13
3.2.16	Process Startup Fails Due to Unhandled Errors in Systemwide OpenVMS Login Procedure	3-14
3.2.17	DBSRC_FILE Service Attribute is Not Supported on OpenVMS.....	3-14
3.2.18	Installation Clarification: How to Determine Ports to Which to Direct Clients..	3-15
3.2.19	How to Re-create Your RMU_DISP Dispatcher and RMU_SERVICE Service....	3-17
3.3	Oracle SQL/Services V7.1 Client Known Problems and Restrictions.....	3-19
3.3.1	Oracle SQL/Services OpenVMS Client is Now Compiled With DEC C	3-19
3.3.2	Use a Jacket Header File When Calling the Oracle SQL/Services API From C++.....	3-19
3.3.3	Client Enters Nonrecoverable Error State if a Statement With No Parameter Markers is Executed Using Batched Execution	3-20
3.3.4	Incorrect Error Message is Returned if a Client Cancels Batched Execution	3-20
3.3.5	PATHWORKS for DOS and Windows 95	3-21
3.3.6	Client Disconnect Does Not Abort Running Transaction For Transaction Reusable Services	3-21
3.3.7	Applications Using Clients Shipped With Oracle SQL/Services V3.1 or Earlier Are Not Supported, But Are Not Rejected	3-22
3.3.8	Repeat Count on SQLSRV_FETCH_MANY Must be Less Than or Equal to 65535	3-22
3.3.9	SQSAPIW.INI and QSAPI32.INI Example is Misleading	3-22
3.4	Other Oracle SQL/Services Server Known Problems and Restrictions.....	3-23
3.5	Other Oracle SQL/Services Client Known Problems and Restrictions	3-25
3.6	Oracle SQL/Services Documentation Errors or Omissions.....	3-26

4 SQL*Net for Rdb: Release Notes

4.1	Software Requirements	4-1
4.2	Installing SQL*Net for Rdb.....	4-1
4.2.1	Changes to the Installation Procedure	4-2
4.2.2	Installation Verification.....	4-3

4.2.3	Problem Reporting	4-3
4.3	Configuring Sample Files	4-3
4.3.1	Configuring LISTENER.ORA	4-3
4.3.2	Configuring TNSNAMES.ORA.....	4-4
4.3.3	Configuring Optional Files	4-7
4.4	Upgrading Oracle Rdb Release 7.0 Database to Oracle Rdb Release 7.1.0.....	4-8
4.5	Script Names	4-8
4.6	New and Changed Features.....	4-9
4.6.1	Emulate Oracle V8.0.5.1.....	4-9
4.6.2	New Logical Names.....	4-9
4.6.3	New Command File RDB_NATCONN.COM	4-10
4.6.4	Displaying Oracle Messages.....	4-11
4.6.5	Support for Hot Standby.....	4-11
4.6.6	SQL*Net for Rdb Product.....	4-11
4.6.7	Compatibility with Oracle Forms	4-11
4.7	Known Problems and Restrictions.....	4-12
4.7.1	Restrictions and Limitations	4-12
4.7.2	SQL Functions Return Incorrect User Information	4-13
4.7.3	Alterations to Dispatcher Network Port Resets Protocol to SQLSRV	4-14
4.7.4	Problem With RMU Dispatcher	4-15
4.7.5	SQL*Plus Release 3.3: Error Accessing Package DBMS_APPLICATION_INFO	4-16
4.7.6	Oracle Reports: Problem with Rdb Accepting a Built-In Function.....	4-16
4.7.7	OCI: Error When Attempting to Fetch a Binary ROWID.....	4-17
4.7.8	OCI: Error When Program Linked in Nondeferrable Mode.....	4-17
4.7.9	Stored Procedures with Bind Variables	4-18
4.8	Software Errors Fixed	4-18
4.8.1	Environment Problems Reported in the OCI Dispatcher Log File Fixed.....	4-18
4.8.2	Access Violations Fixed.....	4-19
4.8.3	Failure While Holding a Mutex During Executor Cleanup	4-20
4.8.4	Using Bind Variable Twice in a Query.....	4-20
4.8.5	ANSI_DATE and SYSDATE Functions Work Correctly	4-20
4.8.6	Wrong Error Returned When Inserting a Duplicate Value into a Unique Index.....	4-20
4.8.7	SYSDATE Function is Available from a Dblink Connection	4-21
4.8.8	Embedded SQL Program Using SET TRANSACTION Works Correctly.....	4-21

4.8.9	ALL_TAB_COLUMNS and USER_TAB_COLUMNS Views Return Correct Scale and Precision.....	4-21
4.9	Documentation Changes.....	4-21
4.9.1	Guide to SQL*Net for Rdb.....	4-21
4.9.1.1	Changes to Database Privileges.....	4-21
4.9.1.2	New Scripts to Mimic Oracle Public Packages.....	4-22
4.9.1.3	Encrypted Passwords.....	4-22
4.9.1.4	Correction to Character Set for DEC-Kanji.....	4-23

List of Examples

4-1	Specifying the OCI Protocol with SQLSRV_MANAGE71	4-14
4-2	Errors Logged to the Dispatcher Log File	4-14
4-3	Errors Returned When Accessing DBMS_APPLICATION_INFO	4-16
4-4	Stored Function Module Workaround	4-17

List of Tables

1-1	Oracle SQL/Services on OpenVMS Alpha Client/Network Support	1-3
4-1	Script Names	4-9
4-2	SQL*Net for Rdb Restrictions and Limitations.....	4-12

Send Us Your Comments

Oracle Rdb Oracle SQL/Services Release 7.1.5 Release Notes, Release 7.1.0 for OpenVMS

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: nedc-doc_us@oracle.com
- FAX: 603.897.3825 Attn: Oracle Rdb
- Postal service:
Oracle Corporation
Oracle Rdb Documentation
One Oracle Drive
Nashua, NH 03062-2804
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Oracle SQL/Services software is a client/server component of Oracle Rdb. Oracle SQL/Services lets you develop client application programs on a variety of desktop and mainframe systems so that you can access Oracle Rdb databases as well as other databases supported by the Oracle Rdb implementation of the SQL standard.

Oracle ODBC Driver for Rdb release 3.0 is available (in a separate kit on a separate CD-ROM) for the following client operating systems: Windows 95, Windows 98, Windows 2000, Windows NT Intel. The Oracle ODBC Driver for Rdb allows ODBC applications on these clients read and write access to Oracle Rdb databases using TCP/IP, DECnet, and SQL*Net (for MS Windows clients only).

This manual describes new and changed features; problems fixed in this release; and current problems, restrictions, and other notes.

Intended Audience

These release notes are intended for all users of Oracle SQL/Services and SQL*Net for Rdb and should be read to supplement information contained in the *Oracle SQL/Services Release 7.1.5 Installation Guide*, the *Oracle SQL/Services Server Release 7.1.5 Configuration Guide*, and the *Guide to Using the Oracle SQL/Services Client API*.

To get the most out of this manual, you should be familiar with Oracle SQL/Services, data processing procedures, and basic database management concepts and terminology.

Operating System Information

Information about the versions of the operating system and related software that are compatible with this version of Oracle SQL/Services and SQL*Net for Rdb is included in these release notes (see [Section 1.1.2](#)).

Structure

This manual contains four chapters

- Chapter 1** Describes the new and changed features of Oracle SQL/Services release 7.1.5.
- Chapter 2** Describes known software errors in versions prior to release 7.1.5 of Oracle SQL/Services that were fixed in release 7.1.5.
- Chapter 3** Describes problems, restrictions, and workarounds known to exist in Oracle SQL/Services release 7.1.5.
- Chapter 4** Describes release notes that pertain to SQL*Net for Rdb release 7.1.5.

Related Manuals

For more information on Oracle Rdb and Oracle SQL/Services, see the following manuals in this documentation set, especially the following:

- *Oracle SQL/Services Release 7.1.5 Installation Guide*
- *Oracle SQL/Services Server Release 7.1.5 Configuration Guide*
- *Guide to Using the Oracle SQL/Services Client API*

Oracle Rdb Release Notes for release 7.1.0 provides descriptions of, and ordering information for all manuals in the Oracle Rdb documentation set.

The *Oracle SQL/Services Release 7.1.5 Release Notes* and the *Oracle Rdb Release Notes* for release 7.1.0 are provided only as part of the software kit. PostScript (.ps), Adobe Portable Document Format (.pdf), and text (.release_notes) files for the release notes are available in SYSSHELP.

All other books are available in html, pdf, and printed form.

Conventions

In this manual, Oracle Rdb refers to Oracle Rdb for OpenVMS Alpha software. Release 7.1.0 of Oracle Rdb software is often referred to as V7.1.

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, in general referred to as the ANSI/ISO SQL standard or SQL:1999. See the *Oracle Rdb Release Notes* for additional information about this SQL standard.

Oracle ODBC Driver for Rdb software is referred to as the ODBC driver.

OpenVMS means the OpenVMS Alpha operating system.

In examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the Return key at the end of a line of input.

Often in examples the prompts are not shown. Generally, they are shown where it is important to depict an interactive sequence exactly; otherwise, they are omitted in order to focus full attention on the statements or commands themselves.

The following conventions are also used in this manual:

[]	In text, brackets enclose optional information from which you can choose to use or not.
\$	The dollar sign represents the DIGITAL Command Language prompt in OpenVMS.
>	The right angle bracket represents the MS-DOS command prompt. This symbol indicates that the MS-DOS command language interpreter is ready for input.
boldface text	Boldface type in text indicates a term defined in the text.

Oracle SQL/Services: New and Changed Features

This chapter describes the new features and technical changes to Oracle SQL/Services release 7.1.5. In addition, it describes obsolete routines, structures, and features, and provides a summary of additions and changes to the documentation.

1.1 Installing Oracle SQL/Services Release 7.1.5

Refer to the *Oracle SQL/Services Release 7.1.5 Installation Guide* for installation information.

[Section 1.1.2](#) briefly highlights the major changes to the Oracle SQL/Services installation procedure, which are described in detail in the *Oracle SQL/Services Release 7.1.5 Installation Guide*.

1.1.1 Save-Set Names

The Oracle SQL/Services V7.1.5 installation kit contains the following files:

- SQLSRVAMVE071.A – Oracle SQL/Services for OpenVMS Alpha
- SQLSRVAMVE071.B – SQL*Net for Rdb for OpenVMS Alpha
- SQSCLIA071.A – Oracle SQL/Services client kit for OpenVMS Alpha

The minimum supported version of Oracle Rdb and Oracle SQL/Services client software is V6.1.

1.1.2 Software Installation Requirement and Compatibility Information

The following information describes the Oracle SQL/Services hardware and software requirements and compatibility information for installing Oracle SQL/Services release 7.1.5.

Hardware Requirements

Oracle SQL/Services server platforms require hardware configurations and Ethernet LAN connectivity supported by the prerequisite software listed in [Software Requirements](#).

Optional Hardware

Oracle ODBC Driver for Rdb and Oracle SQL/Services client platforms require hardware configurations and Ethernet LAN connectivity supported by the prerequisite software listed in [Software Requirements](#).

Software Requirements

The following software is required for the Oracle SQL/Services OpenVMS Alpha server.

- OpenVMS Alpha operating system V7.1 or higher
- Oracle Rdb for OpenVMS Alpha V7.0 or higher
- One of the following network transport options:
 - DECnet transport - DECnet for OpenVMS Alpha, V7.1 or higher
 - TCP/IP transport - DEC TCP/IP Services for OpenVMS Alpha, V3.3 or other DEC TCP/IP Services for OpenVMS compliant transport on the host system.

Note: TCP/IP Services for OpenVMS Alpha V3.3 has been fully tested with Oracle SQL/Services. Other transports that comply with the DEC TCP/IP Services for the OpenVMS interface may function correctly but have not been fully tested by Oracle Corporation.

- SQL*Net transport - Oracle SQL*Net Server, V2.1.5

Table 1-1 lists the desktop clients and network transports supported by Oracle SQL/Services on the OpenVMS Alpha platform.

Table 1–1 Oracle SQL/Services on OpenVMS Alpha Client/Network Support

Desktop	Client Transport Support		
Client Platform	DECnet	TCP/IP	SQL*Net
Oracle ODBC Driver for Rdb Clients			
Windows NT Intel	X	X	--
Windows 95/Windows 98/ Windows 2000	X	X	--
Oracle SQL/Services Clients			
Windows NT Intel	X	X	--
Windows 95/Windows 98/ Windows 2000	X	X	--
Tru64 UNIX	X	X	X
OpenVMS VAX	X	X	X
OpenVMS Alpha	X	X	X

SQL*Net for Rdb Software Requirements

SQL*Net for Rdb Release 7.1.5 requires Version 7.1 or higher of the OpenVMS operating system.

Desktop Client Software Requirements

This section describes the software required by each desktop client platform.

Oracle SQL/Services Clients

The following section describes the software required for each Oracle SQL/Services client and the supported transport.

Oracle SQL/Services Windows 2000 Client (TCP/IP Transport):

- Microsoft Windows 2000 operating system, V4.0

Oracle SQL/Services Windows 98 Client (TCP/IP Transport):

- Microsoft Windows 98 operating system, V4.10.1998

Oracle SQL/Services Windows NT Intel Client (TCP/IP Transport):

- Microsoft Windows NT operating system, V4.0

Oracle SQL/Services Windows NT Intel Client (DECnet Transport):

- Microsoft Windows NT operating system, V4.0
- PATHWORKS for Windows NT, V4.1B

Windows 95 client (TCP/IP Transport):

- Microsoft Windows 95 operating system, V4.00.950

Windows 95 client (DECnet Transport):

- Microsoft Windows 95 operating system, V4.00.950
- PATHWORKS for DOS and Windows, Windows 95 V1.0A

Oracle SQL/Services Tru64 UNIX client (DECnet or TCP/IP Transports):

- Tru64 UNIX operating system, V3.2 or V4.0
- DECnet/OSI, V2.0, for Tru64 UNIX (required for DECnet applications only)

1.2 Documentation

Documentation for Oracle SQL/Services and SQL*Net for Rdb is available in Hyper Text Markup Language (HTML) and in Adobe Acrobat (PDF) formats. You will find all of this documentation and much more on the Oracle Rdb Documentation CD-ROM included with this kit.

Our intent is to provide you with documentation in the most convenient and useful media available. To this end, we welcome your comments.

1.3 Summary of Oracle SQL/Services New and Changed Features for Release 7.1.5

The following sections summarize and describe new and changed features for Oracle SQL/Services release 7.1.5.

New and Changed Server Features

This section describes new and changed server features.

- The following commands are changed:
 - ALTER DISPATCHER and CREATE DISPATCHER
 - * New argument: PROTOCOL [NATIVE | OCI | SQLSERVICES]
 - ALTER SERVICE and CREATE SERVICE

- * New argument: PROTOCOL [OCI | RMU | SQLSERVICES]
- * New argument: DEFAULT_CONNECT_PASSWORD <password-string>
- * New argument: OWNER_PASSWORD <password-string>
- GRANT USE ON SERVICE
 - * New keyword: GROUP[S] <group-name>
This new keyword appears in the <grant-element> syntax and supports granting the USE privilege descriptor to a group and permit access to the specified service by users who are members of that group.
- REVOKE USE ON SERVICE
 - * New keyword: GROUP[S] <group-name>
This new keyword appears in the <grant-element> syntax and supports revoking the USE privilege descriptor from a group to remove access to the specified service by users who are members of that group.
- SHOW

All SHOW commands have been updated to display the new arguments and keywords that have been added to each command.

New and Changed Client Features

The following describes new and changed client features:

- **Deprecated clients no longer ship with Oracle SQL/Services**
All deprecated Oracle SQL/Services clients no longer ship on the Oracle SQL/Services software kit. See [Section 1.4](#) for more information.
- **Windows 98 and Windows 2000**
Windows 98 and Windows 2000 are now supported operating systems for Oracle SQL/Services release 7.1.5.

New and Changed Server Management Features

The following describes new and changed server management features:

- **Windows 98, Windows 2000, and Windows ME**
Windows 98, Windows 2000, and Windows ME are now supported operating systems for Oracle SQL/Services Manager.

New and Changed General Features

The following new logical names have been added:

- **SQLSRV_DISP_LOGPATH**

Allows you to specify the location of the dispatcher log directory as the following example shows:

```
$ DEFINE/SYSTEM/EXEC SQLSRV_DISP_LOGPATH DKA100:[USER1.LOG]
```

- **SQLSRV_DISP_DUMPPATH**

Allows you to specify the location of the dump file directory as the following example shows:

```
$ DEFINE/SYSTEM/EXEC SQLSRV_DISP_DUMPPATH DKA100:[USER1.DUMP]
```

Both of these logical names have to be defined as system logical names. If you do not define the `SQLSRV_DISP_LOGPATH` logical name or the `SQLSRV_DISP_DUMPPATH` logical name, the default directory for dispatcher log files and dump files is the `SYSSMANAGER` directory.

Once you define either the `SQLSRV_DISP_LOGPATH` or `SQLSRV_DISP_DUMPPATH` logical name, you have to restart the dispatcher.

New and Changed Features for SQL*Net for Rdb

The following describes new and changed features for SQL*Net for Rdb (see [Chapter 4](#) for details):

- The following logical names have been added:
 - `SQLNET_RECO_USER`

Allows the Oracle transaction manager to reconnect to the Oracle Rdb database to verify the results of a transaction in the case of a system failure.
 - `SQLNET_BLOB`

Allows you to use segmented strings as Oracle LOBs.
- The `RDB_NATCONN.COM` command file has been added which can be used to add users with their encrypted passwords to an Oracle Rdb database.

1.4 Deprecated Features

The following Oracle SQL/Services features are deprecated. These features are no longer described in the main body of the *Guide to Using the Oracle SQL/Services*

Client API and *Oracle SQL/Services Release 7.1.5 Installation Guide* and are described in detail only in an appendix in the *Guide to Using the Oracle SQL/Services Client API*. Oracle will not enhance features that are deprecated. Oracle may announce in a future version of Oracle SQL/Services that these deprecated features are obsolete and can no longer be used. Therefore, applications that use these features should be modified accordingly, when possible.

- **Macintosh for MPW API**

The Macintosh for MPW client API is deprecated in Oracle SQL/Services release 7.1.5, is frozen at the V7.0 level, and will not be enhanced in future releases. The Macintosh for MPW client API was supplied with the V7.0 kit, but is no longer supplied with the release 7.1.5 kit.

- **Macintosh for THINK C Client API**

The Macintosh for THINK C client API is deprecated in Oracle SQL/Services release 7.1.5, is frozen at the V7.0 level, and will not be enhanced in future releases. The Macintosh for THINK C client API was supplied with the V7.0 kit, but is no longer supplied with the release 7.1.5 kit.

- **Microsoft Windows 3.1 client API**

The Microsoft Windows 3.1 client API is deprecated in Oracle SQL/Services release 7.1.5, is frozen at the V7.0 level, and will not be enhanced in future releases. The Microsoft Windows 3.1 client API was supplied with the V7.0 kit, but is no longer supplied with the release 7.1.5 kit.

- **Solaris client API**

The Solaris client API is deprecated in Oracle SQL/Services release 7.1.5, is frozen at the V7.0 level, and will not be enhanced in future releases. The Solaris client API was supplied with the V7.0 kit, but is no longer supplied with the release 7.1.5 kit.

- **MS-DOS large memory model client API**

The MS-DOS large memory model client API is deprecated in Oracle SQL/Services V7.0, is frozen at the V6.1 level, and will not be enhanced in future releases. The MS-DOS large memory model client API was supplied with the V6.1 kit. It is not supplied with the release 7.1.5 kit.

- **MS-DOS medium memory model client API**

The MS-DOS medium memory model client API was deprecated in Oracle SQL/Services V5.1, is frozen at the V4.2 level, and will not be enhanced in

future releases. The MS-DOS medium memory model client API was supplied with the V5.1 kit. It is not supplied with the V7.0 or V7.1.5 kits.

- **ULTRIX for VAX client API**

The ULTRIX for VAX client API was deprecated in Oracle SQL/Services V5.1, is frozen at the V4.2 level, and will not be enhanced in future releases. The ULTRIX for VAX client API was supplied with the V5.1 kit. It is not supplied with the V7.0 or V7.1.5 kits.

- **ULTRIX for RISC client API**

The ULTRIX for RISC client API is deprecated in Oracle SQL/Services V7.0, is frozen at the V6.1 level, and will not be enhanced in future releases. The ULTRIX for RISC client API was supplied with the V6.1 kit. It is not supplied with the V7.0 or V7.1.5 kits.

- **SunOS client API**

The SunOS client API is deprecated in Oracle SQL/Services V7.0, is frozen at the V6.1 level, and will not be enhanced in future releases. The SunOS client API was supplied with the V6.1 kit. It is not supplied with the V7.0 or V7.1.5 kits.

- **OS/2 client API**

The OS/2 client API was deprecated in Oracle SQL/Services V5.1, is frozen at the V4.2 level, and will not be enhanced in future releases. The V4.2 OS/2 client API was supplied with the V5.1 kit. It is not supplied with the V7.0 or V7.1.5 kits.

- **VAX format of all API routines**

The VAX format for all API routines was deprecated in Oracle SQL/Services V5.0.

- **Association routines**

- **INFO_TYPE** parameter value, **SQLSRV_INFO_DB_CLASS**, of the **SQLSRV_GET_ASSOCIATE_INFO** routine

The **INFO_TYPE** parameter value, **SQLSRV_INFO_DB_CLASS**, of the **SQLSRV_GET_ASSOCIATE_INFO** routine is deprecated in V7.0. However, this parameter value will continue to work for Oracle SQL/Services V7.0. The **SQLSRV_INFO_DB_CLASS** parameter value describes a flag that returns true if the client is connected to a database service; the value is returned as a longword. This is useful to determine whether an attach is required after the association.

Oracle Corporation recommends that you use the new `INFO_TYPE` parameter value `SQLSRV_INFO_SERVICE_ATTRS`, which returns more information about the service.

- `SQLSRV_SET_SERVER_CLASS`

The `SQLSRV_SET_SERVER_CLASS` routine was deprecated in Oracle SQL/Services V6.1. Oracle Corporation recommends that you use the `CLASS_NAME` field in the association structure as the method of choosing a server class because this method works for multiassociation applications.

- `SQLSRV_SET_TRANSPORT_TYPE`

The `SQLSRV_SET_TRANSPORT_TYPE` routine was deprecated in Oracle SQL/Services V6.1. Oracle Corporation recommends that you use the `XPTTYP` field in the association structure to choose a transport because this method works for multiassociation applications.

- SQL statement routine

- `SQLSRV_EXECUTE` - Execute prepared statement

The routine `SQLSRV_EXECUTE` was deprecated in Oracle SQL/Services V7.0. Oracle Corporation recommends that you code your applications using the `SQLSRV_EXECUTE_IN_OUT` routine.

- Functional interface routines

- `SQLSRV_SQLDA_MAP_DATA` - Return column information

The routine `SQLSRV_SQLDA_MAP_DATA` was deprecated in Oracle SQL/Services V5.0. Oracle Corporation recommends that you code your applications using the `SQLSRV_SQLDA_REF_DATA` routine.

- `SQLSRV_SQLDA_UNMAP_DATA` - Free resources

The routine `SQLSRV_SQLDA_UNMAP_DATA` was deprecated in Oracle SQL/Services V5.0. Oracle Corporation recommends that you code your applications using the `SQLSRV_SQLDA_UNREF_DATA` or `SQLSRV_SQLDA2_UNREF_DATA` routine.

- `SQLSRV_SQLCA_NUM_BATCH_ROWS` - Return number of rows processed in a batch operation

Previous versions of the Guide to Using the Oracle SQL/Services Client API described a routine named `SQLSRV_SQLCA_NUM_BATCH_ROWS` as returning the number of rows processed in a batched execution operation from the `SQLERRD[1]` field of the `SQLCA`. The Oracle SQL/Services server

does not maintain or return a count of the rows processed; therefore, this routine always returned 0. This routine is therefore deprecated in Oracle SQL/Services V7.0. Oracle Corporation recommends that you use the `SQLSRV_SQLCA_COUNT` client API routine to retrieve the number of rows processed in a batched execution operation.

1.5 Obsolete Features

An obsolete feature is one that is no longer supported and that was described as a deprecated feature in a previous release. These features no longer work:

- Association routines
 - `SQLSRV_GET_ENVIRONMENT` - Return environment variable values
The `SQLSRV_GET_ENVIRONMENT` routine was obsolete beginning with V6.1. This routine no longer exists.
 - `SQLSRV_SET_ENVIRONMENT` - Set environment variable values
The `SQLSRV_SET_ENVIRONMENT` routine was obsolete beginning with V6.1. This routine no longer exists.
- `SQLSRV_SET_FILTER` - Define filter for result table
The `SQLSRV_SET_FILTER` was obsolete beginning with Oracle SQL/Services V4.2. This routine no longer exists.
- Structures
 - `SQLSRV_ENV_STR` - Environment variable structure
The `SQLSRV_ENV_STR` structure was obsolete beginning with V6.1. This structure no longer exists.
- Local Mode
Local mode was obsolete beginning with Oracle SQL/Services V6.1. The local mode flag is ignored beginning with V6.1 if it is set in the association structure.

Oracle SQL/Services: Software Errors Fixed

This chapter describes problems with previous versions of the Oracle SQL/Services software that are fixed.

2.1 Oracle SQL/Services Errors Fixed Since Release 7.0

The following known problems found in the Oracle SQL/Services OpenVMS server in previous releases were fixed since the release of V7.0.

2.1.1 Existing Configuration Files Are Preserved During an Installation

Prior to release 7.1.5, when you installed a new Oracle SQL/Services kit, the existing Oracle SQL/Services configuration files were deleted. For example, if you installed the Oracle SQL/Services kit, the new kit deleted the existing configuration file, and then created a default Oracle SQL/Services configuration file in its place. However, if you have an existing Oracle SQL/Services configuration file and you install a release 7.1.5 kit, the existing configuration files are no longer deleted.

Beginning with release 7.1.5, a new Oracle SQL/Services installation no longer deletes existing configurations. If you install the kit and a configuration file already exists for the installation, the existing configuration file is preserved. In this case, installing a new kit does not result in the creation of a default configuration. Thus, the following questions are *not* displayed as a part of the installation procedure:

```
What version of SQL should the GENERIC service specify?  
Do you want to use the default Oracle SQL/Services network ports [NO]
```

Oracle Corporation still recommends that you copy the configuration file (SYSSMANAGER:SQLSRV_CONFIG_FILE*nn*.DAT) to have as a backup before installing a new release of Oracle SQL/Services.

2.1.2 Oracle SQL/Services Management Utility Caused a Bugcheck to Occur

Earlier versions of the Oracle SQL/Services V7.0 server would sometimes bugcheck when managed by an Oracle SQL/Services management utility from a higher version. If the newer Oracle SQL/Services management utility sent a message or tag that was not understood by the Oracle SQL/Services V7.0 server, the V7.0 server would bugcheck.

This problem was fixed in V7.0.1.

2.1.3 Oracle SQL/Services Dispatcher Failed to Completely Shut Down When There Were Clients Connected

If clients were still actively connected to a dispatcher while it was being shut down, the dispatcher failed to shut down completely. The dispatcher process terminated abnormally and connected clients were not disconnected. In the dispatcher log file, you would see the DBS-F-BUGCHECK and SQLSRV-E-INTERNAL errors being logged. In addition, the SHOW CLIENT command of the Oracle SQL/Services system management utility showed clients in the CANCELING state.

This problem was fixed in V7.0 ECO01. The Oracle SQL/Services dispatcher is now able to shut down completely when there are connected clients.

2.1.4 Problems Running Oracle SQL/Services V7.0 Multiversion with a Higher Version of Oracle SQL/Services Standard

There were a number of problems in Oracle SQL/Services V7.0 that would prevent a multiversion installation from running concurrently with a standard installation of a higher version of Oracle SQL/Services.

This problem was fixed in V7.0 ECO01.

2.1.5 Oracle SQL/Services Executor Incorrectly Returned Multi-Octet Character Set Data from Columns of Type NATIONAL CHARACTER

A problem was introduced into the Oracle SQL/Services V7.0 executor whereby it used the SQLLEN value to determine how much data to return to a client from a

column of type CHARACTER or NATIONAL CHARACTER. This resulted in multi-octet character set data being truncated when returned to a client using SQLDA2.

This problem was fixed in V7.0 ECO01. Oracle SQL/Services correctly uses the SQLOCTET_LEN value.

2.1.6 Multiversion Installations Erroneously Specified a Version Number

SQLSRV\$STARTUP.COM and SQLSRV\$SHUTDOWN.COM erroneously specified a version number for multiversion installations when installing and removing the SQLSRV\$API image.

This problem was fixed in V7.0 ECO01.

2.1.7 Executors Failed During Startup with RMS-E-PRV Errors

In prior versions of Oracle SQL/Services, an RDB\$SETVER or SQL\$SETVER procedure invoked in the OpenVMS systemwide login procedure caused Oracle SQL/Services executors to fail during startup with RMS-E-PRV errors.

The RDB\$SETVER and SQL\$SETVER procedures attempt to create temporary files in the sys\$scratch directory (if defined), or in the default directory. In prior versions of Oracle SQL/Services, the initial default directory for an executor process was the default directory of the creating process (usually SYSS\$MANAGER). The default directory is changed to the service owner account default directory in a process-specific command procedure, but this command procedure executes after the systemwide login procedure is executed. Since executors usually do not have write access to the SYSS\$MANAGER directory, executors would fail during startup with RDB-E-PRV when RDB\$SETVER attempted to create a temporary file.

This problem was fixed in V7.0 ECO01. In Oracle SQL/Services V7.0.1, the default directory of the creating process is set to the default directory of the service owner account before an executor is created. Since the executor inherits its default directory from its creator, the default directory for executors is now correctly set to the service owner account default directory before the systemwide login procedure is executed.

2.1.8 Problem Fetching More Than 65535 Rows in fetch_many Mode

In versions of Oracle SQL/Services prior to V7.0.1, a fetch_many with a repeat count of 0 (fetch all rows) would fail during the fetch of the 65536th row.

The server ends the fetch_many stream when the current row number equals the fetch_many repeat count. In versions of Oracle SQL/Services prior to V7.0, the

server stored the current row number in a 16-bit integer. When the 16-bit current row number wrapped back to 0 from 65535, the server thought it was done sending rows and would end the `fetch_many` stream. Since the client had not received an EOS yet, it still expected more rows. This resulted in a -2007 (SQLSRV_PRSEERR) error.

This problem was fixed in V7.0 ECO01. The server now uses a 32-bit integer to store the current row number.

2.1.9 Monitor Process Termination Left Dangling Oracle SQL/Services Processes

If an Oracle SQL/Services monitor process shuts down (normally or abnormally), all other Oracle SQL/Services processes should also terminate. In Oracle SQL/Services V7.0, there were instances when the monitor terminated abnormally, and executors or dispatchers were left dangling.

This problem was fixed in V7.0 ECO01. All known instances of this problem have been corrected in V7.0.1. In addition, `SQLSRVSSHUTDOWN.COM` and `SQLSRV$STARTUP.COM` have been improved to check for and terminate dangling Oracle SQL/Services processes.

2.1.10 All Processes Did Not Explicitly Flush Log File Output

In versions of Oracle SQL/Services prior to V7.0.1, the 1-minute VMS output file timer was used to flush events to component log files. This caused meaningful event log information to be lost when a process was abruptly terminated.

This problem was fixed in V7.0 ECO01. In Oracle SQL/Services V7.0.1, all Oracle SQL/Services processes now explicitly flush log file output after writing an error log event. This forces all events to the log file, even when a process is abruptly terminated.

2.1.11 Installation Did Not Allow DECnet Object Name to Begin with a Numeric Value

When installing Oracle SQL/Services V7.0 multiversion kits, the user is asked whether or not to use default network ports. If the installer chooses to select alternate ports, the installation prompts for port numbers or names. Although it is valid to have a DECnet object name beginning with a numeric value, the Oracle SQL/Services installation would report an error during a create server or create dispatcher operation if a DECnet object name beginning with a numeric value was entered.

This problem was fixed in V7.0 ECO01. The user entered value for a DECnet object is now quoted, allowing object names beginning with numeric values.

2.1.12 Oracle SQL/Services Protocol Message to Monitor Caused a Bugcheck to Occur

In V7.0, if the Oracle SQL/Services system management service received a message it did not understand, the Oracle SQL/Services monitor process would bugcheck and Oracle SQL/Services would shut down. If an Oracle SQL/Services client was incorrectly directed to use an Oracle SQL/Services management port (one specified by a Show Server command) rather than an Oracle SQL/Services dispatcher port (one specified by a Show Dispatcher command), an Oracle SQL/Services protocol message was sent to the Oracle SQL/Services management service.

This problem was fixed in V7.0 ECO01. The Oracle SQL/Services monitor now logs a message and continues running if it receives an Oracle SQL/Services protocol message.

2.1.13 Failure While Holding a Mutex During Executor Cleanup

If a client connection terminated abnormally while executing in an executor process, the executor process is cleaned up and shutdown. In versions of Oracle SQL/Services previous to V7.0.2, this executor cleanup could sometimes cause an executor to shutdown while holding a mutex.

This problem is fixed in V7.0.2.

Oracle SQL/Services: Known Problems

This chapter describes problems and restrictions relating to Oracle SQL/Services release 7.1.5.

3.1 Oracle SQL/Services Manager GUI Known Problems and Restrictions

The following information describes Oracle SQL/Services Manager graphical user interface (GUI) restrictions and known problems:

- Only one copy of the Oracle SQL/Services Manager GUI can run on a PC.
Only one copy of the Oracle SQL/Services Manager GUI can be run on a given PC at any one time.

3.2 Oracle SQL/Services Release 7.1.5 Server Known Problems and Restrictions

The following sections describe Oracle SQL/Services release 7.1.5 server restrictions and known problems.

3.2.1 Impersonation Is Not Supported

PERSONA support is a new feature in Oracle Rdb release 7.1. This feature enables the full impersonation of an OpenVMS user. However, if PERSONA is enabled, you cannot connect to an Oracle Rdb database from Oracle SQL/Services. Impersonation is not supported in Oracle SQL/Services.

3.2.2 Concealed Attributes are Required for Rooted Directory Logicals

When Oracle SQL/Services starts a new monitor, dispatcher or executor process, it uses the SET DEFAULT DCL command to set the initial default disk and directory for the process. In addition, when a new client connects to a universal service with database authorization set to connect user, Oracle SQL/Services calls the SYSSSETDDIR OpenVMS system service to set the default disk and directory for the executor process. To set default to a disk and directory combination that includes a rooted directory logical name, the OpenVMS operating system requires that the rooted directory logical name be defined with the CONCEALED attribute.

Consider a rooted directory logical name ALL_USERS used to reference user directories in the following example:

```
Root top-level user directory:  $1$DKA100:[USERS]
Specific user directory:       $1$DKA100:[USERS.FRED]
```

In this example, the ALL_USERS rooted directory logical name must be defined as follows (the /EXECUTIVE switch may also be used for greater security):

```
$ DEFINE/SYSTEM ALL_USERS $1$DKA100:[USERS.]/TRANSLATION_ATTRIBUTE=CONCEALED
```

The default disk and directory for user FRED can then be specified as follows:

```
ALL_USERS:[FRED]
```

If a rooted directory logical name is not defined with the CONCEALED attribute, then the SET DEFAULT DCL command executed during monitor or dispatcher process creation fail as follows if the monitor is started from an account that specifies the rooted directory logical name. Likewise, the SET DEFAULT DCL command executed during executor process creation will also fail in the same way if the service owner user name account specifies the rooted directory logical name.

```
$ DEFINE SYS$LOGIN ALL_USERS:[FRED]
$ SET DEFAULT SYS$LOGIN
%DCL-W-DIRECT, invalid directory syntax - check brackets and other delimiters
```

In addition, if a rooted directory logical name specified for a client account is not defined with the CONCEALED attribute, then an executor process will bugcheck and exit with the following error message in the executor log if the user connects to a universal service with database authorization set to connect user:

```
-----EVENT BEGIN:  EVENT_LOG at Wed Sep 24 1997
14:05:33.914-----%SQLSRV-I-EVENT_LOG, event logged at line 1636 in file
DBS_PROT_VMS.C;1
Error setting VMS process user name
```

```
%RMS-F-DIR, error in directory name
-----EVENT END : EVENT_LOG at Wed Sep 24 1997 14:05:33.930-----
```

See the *Guide to OpenVMS File Applications* in the OpenVMS documentation set for more information on how to define and use rooted directory logical names.

3.2.3 Problems With Installing Oracle Rdb V6.1 After Installing Oracle SQL/Services V7.0 or Higher

The following problems occur when you install Oracle Rdb V6.1 after installing Oracle SQL/Services V7.0 or higher

- SYSSSHARE:sqlsrv\$mod61.exe or SYSSSHARE:sqlsrv\$mod.exe may be replaced.

Oracle SQL/Services implements database services using a number of new SQL module language procedures. In order to use database services with Oracle Rdb V6.1 databases, the V7.0 or higher installation procedure updates the Oracle SQL/Services SQL module procedure shared image for Oracle Rdb V6.1:

```
SYSSSHARE:sqlsrv$mod61.exe or SYSSSHARE:sqlsrv$mod.exe.
```

If you install Oracle Rdb V6.1 ECO01 or earlier after installing Oracle SQL/Services V7.0 or higher, the SQL module procedure shared image will be overwritten and errors occur.

If Oracle Rdb V6.1 (no ECO) is reinstalled after Oracle SQL/Services V7.0 or higher is installed, the following errors occur when you attempt to start a database service:

```
-----EVENT BEGIN: EVENT_LOG at Tue Jul 23 1996 15:17:47.250-----
%SQLSRV-I-EVENT_LOG, event logged at line 739 in file SRVSQLINIT.C;1
%SQLSRV-E-SHRIMGLOOKUPERR, Error looking up symbol:
SQLSRV$MOD_LWC_CONNECT_IMP in shared image: SYS$COMMON:[SYSLIB]SQLSRV$MOD61.EXE
%SQLSRV-E-ERROR_TEXT, Error text:
%LIB-E-KEYNOTFOU, key not found in tree
%SQLSRV-F-BADSQLSRVINSTAL, Required SQL/Services entry-point not found
%SQLSRV-E-EXECINITFAILED, Executor initialization failed; process terminating
-----EVENT END : EVENT_LOG at Tue Jul 23 1996 15:17:47.275-----
```

If Oracle Rdb V6.1 ECO01 is installed after Oracle SQL/Services V7.0 or higher is installed, the following errors occur when you attempt to connect to a database service:

```
-----EVENT BEGIN: EVENT_LOG at Wed Jul 24 1996 15:46:55.463-----
```

```
%SQLSRV-I-EVENT_LOG, event logged at line 2217 in file SRVEXE.C;1
%SQLSRV-E-LWCONNERRUSER, Error creating a SQL connect for username: USERNAME
%SQL-F-ARGCOUNT, Procedure SQLSRV$MOD_LWC_CONNECT_IMP expected 5 parameters, was
  passed 4
-----EVENT END : EVENT_LOG at Wed Jul 24 1996 15:46:55.481-----
```

- SYSSSHARE:sql\$setver.com may be replaced.

Oracle SQL/Services uses new logical names defined by SYSSSHARE:SQL\$SETVER.COM to determine the version of SQL being run, and to find the correct Oracle SQL/Services SQL module procedure shared image. If an older version of SQL\$SETVER.COM is found on the system, the Oracle SQL/Services V7.0 or higher installation procedure updates the SQL\$SETVER.COM file.

If you install Rdb V6.1 or any ECO of Oracle Rdb V6.1 after installing Oracle SQL/Services V7.0 or higher, SYSSSHARE:sql\$setver.com may be replaced with an older version and the following errors may occur when a service is started:

```
-----EVENT BEGIN: EVENT_LOG at Thu Jul 25 1996 11:04:36.378-----
%SQLSRV-I-EVENT_LOG, event logged at line 863 in file SRVSQLINIT.C;1
%SQLSRV-W-SQLVERNOTFOUND, Could not determine SQL version; results unpredictable
-----EVENT END : EVENT_LOG at Thu Jul 25 1996 11:04:36.518-----
```

In addition, if the Oracle SQL/Services service sets its SQL version to a version of SQL that was installed as multiversion, the following errors may occur when a service is started:

```
-----EVENT BEGIN: EVENT_LOG at Thu Jul 25 1996 11:04:36.548-----
%SQLSRV-I-EVENT_LOG, event logged at line 739 in file SRVSQLINIT.C;1
%SQLSRV-E-SHRIMGLOOKUPERR, Error looking up symbol: SQLSRV$MODCLSCRS in shared i
  mage: SQLSRV$MOD
%SQLSRV-E-ERROR_TEXT, Error text:
%LIB-E-ACTIMAGE, error activating image PENINE$DKA0:[SYS0.SYSCOMMON.][SYSLIB]SQL
  SRV$MOD.EXE;
-RMS-E-FNF, file not found
%SQLSRV-F-SQLNOTINSTALLED, Specified SQL version is not installed
%SQLSRV-E-EXECINITFAILED, Executor initialization failed; process terminating
-----EVENT END : EVENT_LOG at Thu Jul 25 1996 11:04:36.578-----
```

To correct either of these problems, you can reinstall Oracle SQL/Services V7.0 or higher, or you can extract the correct files from the Oracle SQL/Services V7.0 or higher kit saveset. The following examples illustrate the procedure for extracting files from the Oracle SQL/Services OpenVMS Alpha multiversion kit (the following examples are from a release 7.0 installation):

```

! Extract SQL$SETVER.COM and put it in SYS$COMMON:[SYSLIB]
  $BACKUP/SELECT=(sql$setver.com) -
<dev><dir>SQLSRVAM070.A/SAVE -
  SYS$COMMON:[SYSLIB]*/LO

! Extract SQLSRV$MOD.EXE and put it in SYS$COMMON:[SYSLIB]. Extract this
! file only if Rdb 6.1 or SQL 6.1 is installed standard.
  $BACKUP/SELECT=(sqlsrv$mod.exe) -
<dev><dir>SQLSRVAM070.A/SAVE -
  SYS$COMMON:[SYSLIB]*/LO

! Extract SQLSRV$MOD61.EXE and put it in SYS$COMMON:[SYSLIB]. Extract this
! file if Rdb 6.1 or SQL 6.1 is installed multiversioned.
  $BACKUP/SELECT=(sqlsrv$mod61.exe) -
<dev><dir>SQLSRVAM070.A/SAVE -
  SYS$COMMON:[SYSLIB]*/LO

```

3.2.4 Database Authorization for Database Services

In V7.0, Oracle SQL/Services on OpenVMS systems added the ability to define database services with database access authorization set to the connect user name. In versions of Oracle SQL/Services previous to V7.0, database services always accessed the database using the service owner user name.

For database services, database access using the connect user name is only available for services that attach to Oracle Rdb V6.1 or higher databases. Database services that attach to Oracle Rdb V6.0 or earlier databases must specify database authorization as the service owner for database services. This restriction does not apply to universal services (those services not preattached to a database).

3.2.5 Installation Clarification: How to Determine Ports to Which to Direct Clients

As of Oracle SQL/Services V7.0, it is possible to run multiple versions of Oracle SQL/Services at the same time on the same system. In order to do this, each Oracle SQL/Services server must use unique ports.

Note that each Oracle SQL/Services server uses multiple sets of ports; one set for the server for system management, and one set for each defined dispatcher. A Show Server command displays the set of ports used for system management:

```

SQLSRV> show server;
Server Version:      7.0
Server Platform:    OpenVMS Alpha
Max Shared Mem Size: 2000 Kb

```

```

Config file:          SYS$SYSROOT:[SYSMGR]SQLSRV_CONFIG_FILE70.DAT;1
Log path:            SYS$MANAGER:
Dump path:          SYS$MANAGER:
Proc start time:    <none>
Proc shut time:     <none>
Network Ports:
DECnet object      SQLSRV_SERVER          (State)  (Protocol)
TCP/IP port        2199                    Running  Native
Current shared memory usage:
Allocation unit:    65536 bytes
Total memory:       2031616 bytes ( 31 units)
Free memory:        1835008 bytes ( 28 units)
Partly allocated:   131072 bytes ( 2 units)
Log File:           SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_MON_0070.LOG;
Dump File:          SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_70.DMP;

```

During installation, a `SQLSRV_DISP` dispatcher is defined to handle requests from Oracle SQL/Services clients. These include Oracle ODBC Driver for Rdb clients, Query Performance Tuner, and the Rdb Web Agent. The `SQLSRV_DISP` dispatcher listens on a specific set of network ports shown as follows under Network Ports. It is also defined to listen for Oracle SQL/Services protocol messages shown as follows under Protocol. A Show Dispatcher command displays the set of ports and protocol used for client communication.

```

SQLSRV> show dispatcher SQLSRV_DISP;
Dispatcher SQLSRV_DISP
State:                RUNNING
Autostart:            on
Max connects:         100 clients
Idle User Timeout:   <none>
Max client buffer size: 5000 bytes
Network Ports:
IPX/SPX port         0x84b1          (State)  (Protocol)
DECnet object        81                    Running  SQL/Services
TCP/IP port          118                    Running  SQL/Services
Log File:            SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_DIS00370.LOG;
Dump File:           SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_DIS003.DMP;

```

During installation, an `RMU_DISP` dispatcher is defined to handle requests from RMU clients. The `RMU_DISP` listens on a specific set of network ports shown as follows under Network Ports. It is also defined to listen for Native protocol messages shown as follows under Protocol. Note that RMU clients send Native protocol messages. A Show Dispatcher command displays the set of ports and protocol used for RMU client communication.

```

SQLSRV> show dispatcher RMU_DISP;
Dispatcher RMU_DISP
  State:                                RUNNING
  Autostart:                             on
  Max connects:                           100 clients
  Idle User Timeout:                       <none>
  Max client buffer size:                   5000 bytes
  Network Ports:                           (State)  (Protocol)
    DECnet object RMU_DISP                   Running  Native
    TCP/IP port   1571                       Running  Native
  Log File:                                SYS$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP00870.LOG;
  Dump File:                                SYS$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP008.DMP;

```

The *Oracle SQL/Services Installation Guide* discusses directing Oracle SQL/Services clients to the correct dispatcher for the desired version of Oracle SQL/Services. This guide neglects to mention how to determine the port number to which to direct the client. Oracle SQL/Services clients must communicate with Oracle SQL/Services dispatchers (those defined with Protocol SQL/Services). Do a Show Dispatcher command to find the desired dispatcher, and direct the client to the ports defined for that dispatcher.

3.2.6 How to Recreate Your RMU_DISP Dispatcher and RMU_SERVICE Service

During Oracle SQL/Services installation, an RMU_DISP dispatcher and an RMU_SERVICE service are created. These components listen for and execute requests from RMU clients.

The RMU_DISP dispatcher listens on a specific set of network ports as shown in the following example under Network Ports. RMU clients cannot be configured to use alternate ports, so it is important that the network ports for the RMU_DISP dispatcher are not changed.

The RMU_DISP dispatcher is also defined to listen for Native protocol messages as shown in the following example under Protocol. Note that RMU clients send Native protocol messages. It is important that the Protocol for the RMU_DISP dispatcher remain as Native.

A Show Dispatcher command displays the set of network ports and protocol used for RMU client communication.

```

SQLSRV> show dispatcher RMU_DISP;
Dispatcher RMU_DISP
  State:                                RUNNING
  Autostart:                             on
  Max connects:                           100 clients

```

```

Idle User Timeout:          <none>
Max client buffer size:    5000 bytes
Network Ports:              (State) (Protocol)
  DECnet object  RMU_DISP          Running Native
  TCP/IP port    1571              Running Native
Log File:                   SYS$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP00870.LOG;
Dump File:                  SYS$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP008.DMP;

```

The **RMU_SERVICE** service executes requests from RMU clients. The Service Protocol determines the type of request a service can execute. It is important that the Protocol for the **RMU_SERVICE** service remain as **RMU**.

A **Show Service Full** command displays the protocol used for the RMU service.

```

SQLSRV> show service rmu_service full;
Service RMU_SERVICE
  State:                RUNNING
  Owner:                RMU$SRV
  Protocol:             RMU
  Default Connect Username: <not specified>
  SQL version:          7.0
  Autostart:            on
  Process init:         <not specified>
  Attach:               <not specified>
  Schema:               <not specified>
  Reuse:                SESSION
  Database Authorization: CONNECT USERNAME
  dbsrc file:           <not specified>
  SQL init file:        <not specified>
  Appl Transaction Usage: SERIAL
  Idle User Timeout:    <none>
  Idle Exec Timeout:    120 seconds
  Min Executors:        4
  Max Executors:        100
  Running Executors:    4
  Clients Per Executor: 1
  Active Clients:       0

```

```

Access to service RMU_SERVICE
  Granted to users:
    PUBLIC PRIVILEGED_USER 'RMU$SRV'

```

Oracle SQL/Services did not document the protocol attributes of the **Create** and **Alter Dispatcher** and **Service** commands in V7.0. There is no way to specify the

dispatcher protocol or service protocol using the Oracle SQL/Services Manager GUI in V7.0.

If you lose or alter your RMU_DISP dispatcher or RMU_SERVICE service definitions, they can be recreated by invoking the SYSSMANAGER:SQLSRV_CREATE71.COM command procedure. This procedure invokes the SYSSMANAGER:SQLSRV_CREATE71.SQS system management script to create the RMU_DISP dispatcher and the RMU_SERVICE service as follows:

```
--
-- Create the Oracle RMU dispatcher.
--

create dispatcher rmu_disp
    autostart on
    network_port decnet object rmu_disp protocol native
    network_port tcpip port_id 1571 protocol native;

--
-- Create the Oracle RMU service for 7.0.
--

create service rmu_service
    protocol rmu
    autostart on
    sql version 7.0
    owner 'RMU$SRV'
    database authorization connect username
    min_executors 4
    max_executors 100
    idle_executor_timeout 120;

--
-- Grant access to the Oracle RMU service to all users. This
-- allows any client that supplies a valid user name and
-- password to access this service.
--

grant use on service rmu_service to public;
```

3.2.7 Do Not Kill Oracle SQL/Services Processes

Under certain circumstances, the entire Oracle SQL/Services server shuts down if an Oracle SQL/Services dispatcher or executor is abnormally terminated. You should never use the DCL STOP/ID command on OpenVMS systems to stop an

Oracle SQL/Services dispatcher or executor process. The Oracle SQL/Services system management command SHUTDOWN DISPATCHER or SHUTDOWN SERVICE should be used to stop dispatchers and executors. If an executor does not terminate after issuing the SHUTDOWN SERVICE command, or if you do not want to shut down the entire service, the Oracle SQL/Services system management command, KILL EXECUTOR, should be used instead.

Note that the RMU Close command can have the same effect as the STOP/ID or kill command by terminating Oracle SQL/Services executors attached to the database being closed. Before using the RMU Close command, ensure that no Oracle SQL/Services executors currently have the database open. Any executors you find that do have the database open should be terminated with either the Oracle SQL/Services system management command SHUTDOWN SERVICE or the KILL EXECUTOR command.

3.2.8 Oracle SQL/Services Does Not Support Columns That Have a Negative Scale Factor

In versions of Oracle SQL/Services prior to V7.0, when preparing a statement containing a reference to a column with a negative scale factor, the executor incorrectly calculates the amount of memory required for the SQLDA and SQLDATA structures. This results in the executor bugchecking with the error message "SQL Services Internal Error: 3, %SQLSRV-F-INTERR, Internal Error: -29460".

In Oracle SQL/Services V7.0, the executor incorrectly calculates the amount of memory required for the SQLDATA structure. This results in data from columns with a negative scale factor being truncated. However, in this situation, no error message is written to the executor log file nor is an error returned to the client application.

3.2.9 Executor Bugchecks if a Pre-V7.0 Oracle SQL/Services Client API Sends an Inappropriate Batch-Execute Abort Message

In versions of Oracle SQL/Services prior to V7.0, the Oracle SQL/Services client API sends an inappropriate message to the executor if the client application calls `sqlsrv_execute` with the `SQLSRV_EXE_ABORT` flag, and if writing the `ABORT_MESSAGE` tag completely fills the message buffer. The inappropriate message causes all versions of the Oracle SQL/Services executor to bugcheck. For example, the Oracle SQL/Services V7.0 executor will bugcheck with a message similar to the following:

```

-----EVENT BEGIN:  EVENT_LOG at Tue May 21 1996 18:41:44.323-----
%SQLSRV-I-EVENT_LOG, event logged at line 831 in file SRVUTL.C;1
%SQLSRV-F-INTERR, Internal error -2011 in SQL/Services executor version V7.0-00
at line 1833 in module SRVEXEMAIN
%SQLSRV-E-ERROR_TEXT, Error text: invalid packet sequence number 1, expected 0
-----EVENT END   :  EVENT_LOG at Tue May 21 1996 18:41:44.351-----

```

Note: This problem is fixed for the V7.0 Oracle SQL/Services client API, but not for any pre-V7.0 Oracle SQL/Services client API using an Oracle SQL/Services V7.0 executor.

3.2.10 Matching UIC Required to Use Oracle Rdb V6.1 With a Database Service That Has Database Authorization Set to Connect User Name

If a client application attempts to connect to a database service configured to use Oracle Rdb V6.1 that has database authorization set to connect user name, then the `sqlsrv_associate` call fails with a `SQL_RDBERR (-1)`, "%RDB-F-AUTH_FAIL, authentication failed for user !AC" error if there is no identifier for the user account's UIC or if the name of the UIC identifier does not match the username.

3.2.11 Do Not Shut Down or Restart the SQLSRV_MANAGE System Management Service

If you shut down or restart the `SQLSRV_MANAGE` system management service using either the `SQLSRV_MANAGE` utility or the Oracle SQL/Services Manager GUI, then subsequent attempts to connect to the server are rejected and you render the server unmanageable. If you do accidentally shut down or restart the `SQLSRV_MANAGE` service, then you must find and kill the Oracle SQL/Services monitor process, then restart the server.

3.2.12 Do Not Abbreviate AUTOCONNECT or AUTOSTART Keywords

The `SQLSRV_MANAGE` syntax now includes an `AUTOCONNECT` and an `AUTOSTART` attribute; therefore, the abbreviation "AUTO" is not unique. If the abbreviation "AUTO" is used, a misleading syntax error is generated. For example:

```

SQLSRV> create service v61 auto on sql version 6.1 owner 'sqlsrv$deflt';
syntax error at ''Owner', required parameter not specified'

```

3.2.13 Oracle SQL/Services Management Utilities Do Not Prevent Multiple Dispatchers With the Same Port IDs

Oracle SQL/Services allows you to define multiple dispatchers, each listening on different network ports. Currently, the SQLSRV_MANAGE and the Oracle SQL/Services Manager GUI do not ensure that multiple dispatchers do not use the same port numbers or names. If multiple dispatchers are defined to use the same ports, the second dispatcher to be started fails.

3.2.14 Database Service Attached to a Remote Database Does Not Know if the Database is Closed

It is possible for Oracle SQL/Services database services to be preattached to a remote database. For example, the payroll service defined below attaches to the database "payroll" on node "REMOTE".

```
create service payroll autostart on
reuse session
sql version 7.0
attach 'filename REMOTE::payroll'
owner 'payrollacct'
database authorization service owner
min_executors 5
max_executors 5
```

If the payroll database on node REMOTE is closed, the Oracle SQL/Services payroll service has no way of knowing that the database has been closed. The payroll service continues to run, even though it is no longer attached to the database. The service is useless and must be shut down and restarted after the database is reopened. Any clients attached to the service while it is in this state get a SQLCODE of -1 with the following errors when they attempt to access the database:

```
%RDB-F-IO_ERROR, input or output error
-SYSTEM-F-LINKABORT, network partner aborted logical link
```

All Oracle SQL/Services services that are preattached to a remote database should be shut down before the database is closed. If this is not possible, there is a workaround for database services defined to attach to Oracle Rdb V6.1 or higher databases. Rather than defining session reusable database services, you can define a transaction reusable database service with CLIENTS_PER_EXECUTOR set to 1.

```
create service payroll autostart on
reuse session
sql version 7.0
```

```
attach 'filename REMOTE::payroll'  
owner 'payrollacct'  
database authorization service owner  
min_executors 5  
max_executors 5  
clients_per_executor 1
```

The service definition previously shown gives you essentially the same behavior as the previous session reusable database service. However, Oracle SQL/Services executes a "get diagnostics ? = transaction_active" statement to detect the end of a transaction for transaction reusable services. Because this requires a call to the Oracle Rdb engine, it fails and Oracle SQL/Services bugchecks and shuts down the executor. If this brings the executor count below the MIN_EXECUTORS value defined for the service, the Oracle SQL/Services monitor attempts to create a new executor process. If the monitor fails to start a new executor process after two attempts, it shuts down the service. Note that this workaround generates executor bugcheck dumps that need to be cleaned up.

3.2.15 Log and Bugcheck Dump File Path Server Attributes are Ignored on OpenVMS Systems

Monitor, dispatcher log, and bugcheck dump files are written to the SYSSMANAGER location unless you have defined the SQLSRV_DISP LOGPATH and SQLSRV_DISP_DUMPPATH logical names. If you have defined the SQLSRV_DISP LOGPATH and SQLSRV_DISP_DUMPPATH logical names, the dispatcher log and bugcheck dump file will be written to the specified location.

Executor log files are always written to the default directory of the service owner's account. For a universal service with database authorization set to the connect user name, executor bugcheck dump files are written to the default directory of the connect user name; for all other types of service, the executor bugcheck dump files are written to the default directory of the service owner's account.

Note: The quickest way to locate a bugcheck dump file for a monitor, dispatcher, or executor is to check the monitor log file. The monitor log file describes the location of the dispatcher and executor log files and the location, if any, of any monitor bugcheck dump files. The dispatcher and executor log files describe the location, if any, of any dispatcher and executor bugcheck dump files, respectively.

3.2.16 Process Startup Fails Due to Unhandled Errors in Systemwide OpenVMS Login Procedure

All processes in the Oracle SQL/Services server environment on OpenVMS are created running the SYSSYSTEM:loginout image with a process-specific command procedure as SYS\$INPUT. Because the loginout image is used to create the process, the systemwide login procedure will be executed by the loginout image during process creation. If this procedure fails for some reason, then the Oracle SQL/Services process will fail to start. By default, any DCL command or image that completes with a failure status with a severity level of error or fatal can cause the procedure to fail unless it is handled using the DCL ON or SET NOON commands.

All Oracle SQL/Services processes start by executing the following DCL commands during process creation:

```
$ DELETE/SYMBOL/ALL
$ VRFY_SAVE = F$VERIFY(1)
$ DELETE <disk>:[directory]SQS_<node>_<component>.COM;
$ DEFINE SQS$DBSERVER TRUE
$ DEFINE SYS$LOGIN "<disk>:[directory]"
$ SET DEFAULT SYS$LOGIN
$ DEFINE SYS$SCRATCH "<disk>:[directory]"
```

If an Oracle SQL/Services process fails before executing these commands, please review the systemwide login procedure to determine the reason for the failure.

3.2.17 DBSRC_FILE Service Attribute is Not Supported on OpenVMS

The Oracle SQL/Services DBSRC_FILE service attribute is not currently supported on OpenVMS. To specify values in either the RDB\$CLIENT_DEFAULTS.DAT file or the RDB\$SERVER_DEFAULTS.DAT file, you must define a logical name that specifies the directory that contains one or both of these files. To specify the directory on a per-process basis in the Oracle SQL/Services environment, you must create an executor process initialization command procedure that defines the RDB\$USER_DEFAULTS logical name, then specify the file name of the procedure in the PROCESS_INITIALIZATION service attribute. Alternatively, you can define the RDB\$SYSTEM_DEFAULTS logical as a system logical name or the RDB\$GROUP_DEFAULTS logical name as a group logical name. See the *Oracle Rdb7 SQL Reference Manual* for more information.

3.2.18 Installation Clarification: How to Determine Ports to Which to Direct Clients

Note: Information in this note is also applicable to clients.

As of Oracle SQL/Services V7.0, it is possible to run multiple versions of Oracle SQL/Services at the same time on the same system. In order to do this, each Oracle SQL/Services server must use unique ports.

Note that each Oracle SQL/Services server uses multiple sets of ports; one set for the server for system management, and one set for each defined dispatcher. A Show Server command displays the set of ports used for system management:

```
SQLSRV> show server;
  Server Version:      7.0
  Server Platform:    OpenVMS Alpha
  Max Shared Mem Size: 2000 Kb
  Config file:        SYS$SYSROOT:[SYSMGR]SQLSRV_CONFIG_FILE70.DAT;1
  Log path:           SYS$MANAGER:
  Dump path:          SYS$MANAGER:
  Proc start time:    <none>
  Proc shut time:    <none>
  Network Ports:
    DECnet object    SQLSRV_SERVER          (State) (Protocol)
    TCP/IP port      2199          Running Native
  Current shared memory usage:
  Allocation unit:   65536 bytes
  Total memory:      2031616 bytes ( 31 units)
  Free memory:       1835008 bytes ( 28 units)
  Partly allocated:  131072 bytes ( 2 units)
  Log File:          SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_MON_0070.LOG;
  Dump File:         SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_70.DMP;
```

During installation, a SQLSRV_DISP dispatcher is defined to handle requests from Oracle SQL/Services clients. These include Oracle ODBC Driver for Rdb clients, Query Performance Tuner, and the Rdb Web Agent. The SQLSRV_DISP dispatcher listens on a specific set of network ports shown as follows under Network Ports. It is also defined to listen for Oracle SQL/Services protocol messages shown as follows under Protocol. A Show Dispatcher command displays the set of ports and protocol used for client communication.

```
SQLSRV> show dispatcher SQLSRV_DISP;
Dispatcher SQLSRV_DISP
  State:              RUNNING
```

```

Autostart:                on
Max connects:              100 clients
Idle User Timeout:        <none>
Max client buffer size:   5000 bytes
Network Ports:            (State) (Protocol)
  IPX/SPX port    0x84b1                Running  SQL/Services
  DECnet object  81                      Running  SQL/Services
  TCP/IP port    118                      Running  SQL/Services
Log File:            SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_DIS00370.LOG;
Dump File:           SYS$SYSROOT:[SYSMGR]SQS_ORASQS_SQLSRV_DIS003.DMP;

```

During installation, an RMU_DISP dispatcher is defined to handle requests from RMU clients. The RMU_DISP listens on a specific set of network ports shown as follows under Network Ports. It is also defined to listen for Native protocol messages shown as follows under Protocol. Note that RMU clients send Native protocol messages. A Show Dispatcher command displays the set of ports and protocol used for RMU client communication.

```

SQLSRV> show dispatcher RMU_DISP;
Dispatcher RMU_DISP
  State:                RUNNING
  Autostart:            on
  Max connects:         100 clients
  Idle User Timeout:    <none>
  Max client buffer size: 5000 bytes
  Network Ports:        (State) (Protocol)
  DECnet object  RMU_DISP                Running  Native
  TCP/IP port    1571                      Running  Native
Log File:            SYS$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP00870.LOG;
Dump File:           SYS$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP008.DMP;

```

The Oracle SQL/Services Installation Guide discusses directing Oracle SQL/Services clients to the correct dispatcher for the desired version of Oracle SQL/Services. However, it neglects to mention how to determine the port number to which to direct the client. Oracle SQL/Services clients must communicate with Oracle SQL/Services dispatchers (those defined with Protocol SQL/Services). Do a Show Dispatcher command to find the desired dispatcher, and direct the client to the ports defined for that dispatcher.

3.2.19 How to Re-create Your RMU_DISP Dispatcher and RMU_SERVICE Service

Note: Information in this note is also applicable to Oracle RMU clients.

During Oracle SQL/Services installation, an RMU_DISP dispatcher and an RMU_SERVICE service are created. These components listen for and execute requests from RMU clients.

The RMU_DISP dispatcher listens on a specific set of network ports as shown in the following example under Network Ports. RMU clients cannot be configured to use alternate ports, so it is important that the network ports for the RMU_DISP dispatcher are not changed.

The RMU_DISP dispatcher is also defined to listen for Native protocol messages as shown in the following example under Protocol. Note that RMU clients send Native protocol messages. It is important that the Protocol for the RMU_DISP dispatcher remain as Native.

A Show Dispatcher command displays the set of network ports and protocol used for RMU client communication.

```
SQLSRV> show dispatcher RMU_DISP;
Dispatcher RMU_DISP
  State:                RUNNING
  Autostart:            on
  Max connects:         100 clients
  Idle User Timeout:   <none>
  Max client buffer size: 5000 bytes
  Network Ports:
    DEConn object      RMU_DISP          (State) (Protocol)
    TCP/IP port        1571              Running Native
  Log File:            SYS$$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP00870.LOG;
  Dump File:           SYS$$SYSROOT:[SYSMGR]SQS_ORASQS_RMU_DISP008.DMP;
```

The RMU_SERVICE service executes requests from RMU clients. The Service Protocol determines the type of request a service can execute. It is important that the Protocol for the RMU_SERVICE service remain as RMU.

A Show Service Full command displays the protocol used for the RMU service.

```
SQLSRV> show service rmu_service full;
Service RMU_SERVICE
  State:                RUNNING
```

```
Owner:                RMU$SRV
Protocol:             RMU
Default Connect Username: <not specified>
SQL version:         7.0
Autostart:           on
Process init:        <not specified>
Attach:              <not specified>
Schema:              <not specified>
Reuse:               SESSION
Database Authorization: CONNECT USERNAME
dbsrc file:          <not specified>
SQL init file:       <not specified>
Appl Transaction Usage: SERIAL
Idle User Timeout:   <none>
Idle Exec Timeout:   120 seconds
Min Executors:       4
Max Executors:       100
Running Executors:   4
Clients Per Executor: 1
Active Clients:      0
```

```
Access to service RMU_SERVICE
Granted to users:
    PUBLIC PRIVILEGED_USER 'RMU$SRV'
```

Oracle SQL/Services did not document the protocol attributes of the Create and Alter Dispatcher and Service commands in V7.0. There is no way to specify the dispatcher protocol or service protocol using the Oracle SQL/Services Manager GUI in V7.0.

If you lose or alter your RMU_DISP dispatcher or RMU_SERVICE service definitions, they can be recreated by invoking the SYSSMANAGER:SQLSRV_CREATE[70].COM command procedure. This procedure invokes the SYSSMANAGER:SQLSRV_CREATE[70].SQS system management script to create the RMU_DISP dispatcher and the RMU_SERVICE service as follows:

```
--
-- Create the Oracle RMU dispatcher.
--
create dispatcher rmu_disp
    autostart on
    network_port dectnet object rmu_disp protocol native
    network_port tcpip port_id 1571 protocol native;
--
-- Create the Oracle RMU service for 7.0.
```

```

--
create service rmu_service
  protocol rmu
  autostart on
  sql version 7.0
  owner 'RMU$SRV'
  database authorization connect username
  min_executors 4
  max_executors 100
  idle_executor_timeout 120;
--
-- Grant access to the Oracle RMU service to all users. This
-- allows any client that supplies a valid user name and
-- password to access this service.
--
grant use on service rmu_service to public;

```

3.3 Oracle SQL/Services V7.1 Client Known Problems and Restrictions

The following information describes Oracle SQL/Services V7.1 client known problems and restrictions.

3.3.1 Oracle SQL/Services OpenVMS Client is Now Compiled With DEC C

The Oracle SQL/Services client shared image for OpenVMS is now compiled using DEC C. The options file provided by Oracle SQL/Services for linking client applications has changed. It used to include `SYSSLIBRARY:VAXCTRL$API/SHARE`. It now includes `SYSSLIBRARY:SQLSRV$API/SHARE`.

If you want to relink a client application that was compiled with VAX C, you must create an options file that specifies `SYSSLIBRARY:VAXCTRL/SHARE` and link against this new options file as well as `SYSSLIBRARY:SQLSRV$API.OPT`.

3.3.2 Use a Jacket Header File When Calling the Oracle SQL/Services API From C++

The Oracle SQL/Services header files, `sqlsrv.h`, `sqlsrvca.h`, and `sqlsrvda.h`, do not provide built-in support for use with the C++ programming language. However, by providing a jacket header file, you may call the Oracle SQL/Services API from C++ as you would from C. To include the Oracle SQL/Services header files in a C++ application, create the following header file, called `sqlsrv.hxx`, and `#include` it in your application program:

```
//
// Define VMS if compiling on OpenVMS to pick up the $ versions of
// the service names.
//
#ifdef __VMS
#ifndef VMS
#define VMS
#endif
#endif

//
// Include the headers files using C, not C++. No need to include
// sqlsrvca.h or sqlsrvda.h unless the application directly accesses
// the SQLCA and SQLDA structures.
//
extern "C"
{

#include <sqlsrv.h>
// #include <sqlsrvca.h>
// #include <sqlsrvda.h>

}
```

3.3.3 Client Enters Nonrecoverable Error State if a Statement With No Parameter Markers is Executed Using Batched Execution

If an application executes a prepared statement using the `SQLSRV_EXE_BATCH` flag, but the statement does not contain any parameter markers, the statement is incorrectly executed as if the `SQLSRV_EXE_W_DATA` flag had been specified. That is, the Oracle SQL/Services client API immediately sends an execute request message to the server to execute the statement. At this point, subsequent calls to any API routine, including `sqlsrv_execute_in_out` and `sqlsrv_execute`, all fail with `SQLSRV_INTERR (-2011)` or `SQLSRV_MULTI_ACT (-2016)` errors. Once the client API has entered this error state, only the `sqlsrv_abort` routine functions correctly. Therefore, client applications must not execute SQL statements that do not contain parameter markers using batched execution.

3.3.4 Incorrect Error Message is Returned if a Client Cancels Batched Execution

If an application calls `sqlsrv_execute_in_out` or `sqlsrv_execute` with the execute flag set to `SQLSRV_EXE_WO_DATA` before calling `SQLSRV_EXECUTE_IN_OUT` or

SQLSRV_EXECUTE with the execute flag set to SQLSRV_EXE_BATCH, the client API incorrectly sends an execute request message to the server with no statement ID. Upon receipt of this message, the server returns an SQLSRV_INVSTMID (-2008) error back to the client with the following error message:

```
%SQLSRV-F-INVSTMID, Invalid statement id: 0
```

In this situation, the SQLSRV_INVSTMID error may be ignored.

3.3.5 PATHWORKS for DOS and Windows 95

In order to use the DECnet transport as a transport for Windows 95, use the following procedures:

Install the Windows 95 Release of PATHWORKS for DOS and Windows Version 1.0A.

Version 1.0A is the required version for the DECnet transport. Any Oracle SQL/Services client application that tries running against Version 1.0A will receive the following error:

```
SQLSRV_DLL_ADDR_ERR (-2046).
```

Note: An additional patch is required to the PATHWORKS V1.0A kit. Call the Customer Support Center at Compaq Computer Corporation to obtain this patch. The Oracle ODBC Driver for Rdb will return the following error SQLSRV_DLL_ADDR_ERR (-2046). This means pwsock32.dll, a PATHWORKS dynamic link library, is missing a required routine. The name of the missing PATHWORKS routine is called SktEndNodeEnt.

3.3.6 Client Disconnect Does Not Abort Running Transaction For Transaction Reusable Services

On long-running queries, users may expect that by rebooting the PC the query will be terminated. This is not the case for transaction reusable services. The query will continue until it is ready to send a response to the client. For session reusable services, the query will terminate.

3.3.7 Applications Using Clients Shipped With Oracle SQL/Services V3.1 or Earlier Are Not Supported, But Are Not Rejected

Associations from applications using clients shipped with Oracle SQL/Services Version 3.1 or earlier are not supported but are not rejected by the dispatcher.

3.3.8 Repeat Count on SQLSRV_FETCH_MANY Must be Less Than or Equal to 65535

Because the REPEAT_COUNT parameter to SQLSRV_FETCH_MANY is a 16-bit integer, the maximum number of rows a client can specify on SQLSRV_FETCH_MANY is 65535. If a larger number is specified, no error is detected. Rather, the repeat count wraps around and a smaller repeat count is used. For example, if a repeat count of 65536 is specified, the value in the 16-bit repeat count parameter is 0.

3.3.9 SQSAPIW.INI and SQSAPI32.INI Example is Misleading

The .INI files provided for the Windows platforms specify two commented out sections entitled [RDBSRV]. The first section specifies the transport to be used to communicate with node RDBSRV:

```
; [RDBSRV]
;Transport=DECnet
;Transport=TCP/IP
```

The second section specifies the ports to use to communicate with node RDBSRV:

```
; [RDBSRV]
;TCPIPPortNumber=1040
;DECnetObject=SQLSRV
```

This gives the impression that it is valid to specify two separate sections for each server node name specified in the .INI file. However, it is only valid to have one section per server node name. All server node specific attributes must be specified together in the same section as follows:

```
; [RDBSRV]
;Transport=DECnet
;Transport=TCP/IP
;TCPIPPortNumber=1040
;DECnetObject=SQLSRV
```

Results are unpredictable if two sections are found for a target server node.

3.4 Other Oracle SQL/Services Server Known Problems and Restrictions

The following information describes additional Oracle SQL/Services server for OpenVMS restrictions and known problems:

- LOCALE service attribute is not supported on OpenVMS.

The LOCALE service attribute (CREATE SERVICE and ALTER SERVICE commands) is not supported on OpenVMS.

- Oracle SQL/Services does not support an implicit attach using the SQL\$DATABASE logical name on OpenVMS.

Oracle SQL/Services does not support the use of the SQL\$DATABASE logical name on OpenVMS to implicitly attach to a database. For example, if you define the SQL\$DATABASE logical name, a client application must still issue an explicit SQL ATTACH statement. For example, use ATTACH 'FILENAME SQL\$DATABASE', to attach to the database. If a client application connected to a universal service issues a DML statement before attaching to a database, then the executor will return a status code of -1, with an associated "%SQL-F-NODEFDB, There is no default database" error message.

- Suggested maximum executors of at least 2 for a service.

Many popular desktop tools make two connections to the Oracle SQL/Services server to do their work. For example, MS Access makes one connection initially and returns the list of tables. When the first request to reference a table is made, MS Access makes another connection to the Oracle SQL/Services server. If no executor is available, MS Access returns an error and suggests that you have a problem with your disk or network.

Oracle Corporation recommends that you configure maximum executors of at least 2.

- Decreasing the values for the MIN_EXECUTORS and MAX_EXECUTORS arguments for a transaction reusable service.

When the value for the MIN_EXECUTORS and MAX_EXECUTORS arguments for a transaction reusable service is increased using the ALTER SERVICE command, more executors are made available; however, when the values for the MIN_EXECUTORS and MAX_EXECUTORS arguments are decreased, the values are not changed dynamically. You must perform a SHUTDOWN SERVICE command followed by a START SERVICE command to make fewer executors available.

- Problems that exist for NO_SERVICE and SVCNOTRUN error returns.

Clients may see the NO_SERVICE error returned when the service exists, but has not been started.

Clients may see the SVCNOTRUN (service not running) error when, in fact, the service does not even exist.

- Some error messages are missing object names.

Some error messages from SQLSRV_MANAGE are intended to display the object name that is the source of the error. However, the name is lost and no name is displayed.

- Active dispatcher and executor processes prevent the monitor from starting.

In certain rare error conditions, it is possible for a dispatcher or executor process to remain active after a server is shut down manually or after a server bugchecks and shuts down automatically. If a dispatcher or executor process is active when a server is started, the monitor process fails with the following message in the monitor's bugcheck dump file:

```
%DBS-F-BUGCHECK: in SQS_MON.C:1 at line 1244
Shared memory already exists - monitor startup failed
```

In this situation, look for and stop any active dispatcher or executor processes as follows.

Use the following commands to look for active dispatcher or executor processes:

```
$ SET PROCESS/PRIVILEGE=WORLD
$ SHOW SYSTEM
```

Use the STOP/ID=<pid> command to stop any dispatcher or executor processes that are still active.

- You must call SQLSRV_CLOSE_CURSOR() before using COMMIT or ROLLBACK.

Within SQL, executing a COMMIT or ROLLBACK statement implies that all open cursors are closed unless you are using the Oracle Rdb Hold cursors feature; this assumption is not true for Oracle SQL/Services. Oracle SQL/Services does not parse the SQL statements it passes so it does not know when a commit or rollback operation is executed. Instead, Oracle SQL/Services requires that the SQLSRV_CLOSE_CURSOR() call be issued to release the cursor-related data structures prior to a commit or roll back operation.

To reuse the same cursor name, you must close that cursor before executing a COMMIT or ROLLBACK statement.

- Oracle SQL/Services executors do not execute login.com procedures for clients.

When a client connects to a server, the Oracle SQL/Services executor does not execute the login.com DCL command procedure located in the client user name's default directory. Therefore, client applications should not use logical names defined in LOGIN.COM login procedures. Process logical names for Oracle SQL/Services executors can be defined only by a service's process initialization file.

3.5 Other Oracle SQL/Services Client Known Problems and Restrictions

The following information describes additional Oracle SQL/Services client restrictions and known problems (where a restriction applies to only one platform, an icon denotes to which platform the restriction is applicable; when no platform specific icon is present, the restriction applies to all platforms):

-
- Avoid using cursor names starting with "SQLSRV_".

In designing your applications, you should avoid using cursor names starting with the prefix "SQLSRV_"; this prefix is reserved and used by the Oracle SQL/Services product.
- Oracle SQL/Services compatibility issue with the order of include files.

With V4.1 and higher versions of Oracle SQL/Services, direct access to SQLDA and SQLCA structures is supported but is not recommended by Oracle Corporation. If direct access is used, the order of the Oracle SQL/Services include files must be as follows:

```
#include <sqlsrvca.h>
#include <sqlsrvda.h>
#include <sqlsrv.h>
```

Compile errors will result if the include files are not in this order.
- Allocating space for SQLSRV_VARCHAR and SQLSRV_VARBYTE data types can cause a problem.

Be sure to specify the correct length for the SQLSRV_VARCHAR and SQLSRV_VARBYTE data types in your API applications. Oracle SQL/Services does not issue an error message when the size of the data fields for SQLSRV_VARCHAR and SQLSRV_VARBYTE data types exceeds the size of the SQLLEN field in the SQLDA data structure. See the *Guide to Using the Oracle SQL/Services Client API* for information on allocating space for the SQLSRV_VARBYTE data type and all other data types.

3.6 Oracle SQL/Services Documentation Errors or Omissions

The following information describes Oracle SQL/Services documentation errors or omissions.

- The *Guide to Using the Oracle SQL/Services Client API* does not describe changes to size and format of integer and floating-point data types

Beginning with Oracle SQL/Services V5.1, the size and format of some integer and floating-point data types is changed as follows:

- Trailing zeros occur in fixed-point numeric data types with SCALE FACTOR.

Trailing zeros are now included after the decimal point up to the number of digits specified by the SCALE FACTOR. In versions of Oracle SQL/Services previous to V5.1, at most one trailing zero was included where the value was a whole number.

The following examples illustrate the changes using a field defined as INTEGER(3):

V5.1 and higher	Versions previous to V5.1
1.000	1.0
23.400	23.4
567.890	567.89

- Trailing zeros occur in floating-point data types. Trailing zeros are now included in the fraction, and leading zeros are included in the exponent, up to the maximum precision available, for fields assigned the REAL and DOUBLE PRECISION data types.

Data Type	V5.1 and higher	Versions previous to V5.1

REAL	1.2340000E+01	1.234E+1
DOUBLE PRECISION	5.678900000000000E+001	5.6789E+1

- Size of TINYINT and REAL data types is changed.

The maximum size of the TINYINT and REAL data types is changed to correctly reflect the precision of the respective data types.

The following table shows the maximum lengths of the data types now and in previous versions:

Data type	V5.1 and higher	Versions previous to V5.1
-----	-----	-----
TINYINT	4	6
REAL	15	24

- The *Guide to Using the Oracle SQL/Services Client API* does not describe that the `SQLSRV_ASSOCIATE()` service returns SQL error code -1028 when connecting to a database service if the user has not been granted the right to attach to the database.

When a user connects to a database service, the `SQLSRV_ASSOCIATE()` service completes with the SQL error code -1028, `SQL_NO_PRIV`, if the user has been granted access to the Oracle SQL/Services service, but has not been granted the right to attach to the database. A record of the failure is written to the executor process's log file. Note that the `SQLSRV_ASSOCIATE()` service completes with the Oracle SQL/Services error code -2034, `SQLSRV_GETACCINF` if the user has not been granted access to the Oracle SQL/Services service.

SQL*Net for Rdb: Release Notes

This chapter highlights release notes that pertain to SQL*Net for Rdb release 7.1.5. It contains information about installation, new and changed features, known problems, software fixes, and documentation changes.

4.1 Software Requirements

SQL*Net for Rdb release 7.1.5 requires OpenVMS Alpha Version 7.1 or higher software.

4.2 Installing SQL*Net for Rdb

The installation for SQL*Net for Rdb is part of the installation for Oracle SQL/Services release 7.1.5. Please refer to the following documentation for information on installing SQL*Net for Rdb.

- `SYSSHELP:SQLSRV071.INSTALL_GUIDE`

This document is the *Oracle SQL/Services Release 7.1.5 Installation Guide*. Most of the information required to install SQL*Net for Rdb is in this guide and supersedes the *Guide to SQL*Net for Rdb7*.
- *Guide to SQL*Net for Rdb7*

This document describes the installation and configuration information for SQL*Net for Rdb with Oracle SQL/Services. However, information in the *Oracle SQL/Services Release 7.1.5 Installation Guide* supersedes the information in this guide.

4.2.1 Changes to the Installation Procedure

This section describes changes to the Oracle SQL/Services installation procedure for the SQL*Net for Rdb product. Note the following differences between the Oracle SQL/Services release 7.0 and 7.1.5 installation procedures:

- New save set
The Oracle SQL/Services release 7.1.5 kit includes Save Set B (SQLSRVAMVE071.B) which contains the SQL*Net for Rdb files and images. (The Oracle SQL/Services Release 7.0 kit consisted of a single save set.)
- New question asks for the root directory of your Oracle installation
The Oracle SQL/Services release 7.1.5 installation procedure asks for the root directory of your Oracle installation, if any. Provide the location by taking the actions described in the following table:

If you have an existing Oracle Server installation on this system	Enter the root directory of your Oracle installation. See the translation of the ORA_ROOT logical name to determine this root directory. Do not enter a rooted directory name.
If you have not installed Oracle Server on this system	Enter a carriage return. The Oracle SQL/Services installation procedure creates the following directory: SYS\$COMMON: [SQLSRV71.SQLNET.NETWORK.ADMIN]

The Oracle SQL/Services dispatcher looks for the SQL*Net configuration information and message files in the appropriate directory (as described in the previous table). See *Guide to SQL*Net for Rdb7* for information about distributing the configuration files after installation.

- Default configuration includes the SQL*Net for Rdb components
The Oracle SQL/Services installation procedure creates the following SQL*Net for Rdb samples:
 - Database
 - OCI service
 - OCI dispatcher

4.2.2 Installation Verification

There is no installation verification procedure for SQL*Net for Rdb.

4.2.3 Problem Reporting

If an error occurs while you are using SQL*Net for Rdb and you believe that the error is caused by a problem with this Oracle product, contact your Oracle Corporation support representative for assistance.

When you experience a reproducible problem, it is important to provide as much detailed information as possible. Use the SQL ALTER SESSION LOG FULL statement to collect detailed information about the current SQL*Net for Rdb session. By providing the logged information with your problem report, you supply important data that can help solve the problem. See *Guide to SQL*Net for Rdb7* for more information about using the SQL ALTER SESSION LOG statement.

4.3 Configuring Sample Files

The OpenVMS installation of SQL*Net for Rdb creates sample files to be used as templates for configuring SQL*Net for Rdb. The following sample files are located on the OpenVMS server in SYSSCOMMON:[SQLSRV71.SQLNET.NETWORK.ADMIN]:

- LISTENER.ORA_SAMPLE
- SQLNET.ORA_SAMPLE
- TNSNAMES.ORA_SAMPLE

The two mandatory files which must be configured are the LISTENER.ORA and the TNSNAMES.ORA files.

4.3.1 Configuring LISTENER.ORA

SQL*Net for Rdb requires that an Oracle SQL/Services dispatcher process be running on the OpenVMS server to listen for OCI network traffic. A sample OCI dispatcher object called OCI_DISP is automatically created during the installation of SQL*Net for Rdb. The DBA must create a file called LISTENER.ORA in the directory SYSSCOMMON:[SQLSRV71.SQLNET.NETWORK.ADMIN] before the dispatcher can be started. Any OpenVMS text editor may be used to create this file. Edit the LISTENER.ORA_SAMPLE file and fill in values for the HOST and PORT parameters. OpenVMS node names or TCPIP addresses may be used for the HOST parameter. The default TCPIP Port is 1527. For example:

```
OCI_LISTENER =
(ADDRESS_LIST =
  (ADDRESS =
    (COMMUNITY = TCP_COM.world)
    (PROTOCOL = TCP)
    (Host = ALPHA1)
    (Port = 1527)
  ))
```

Make sure that this file is written as LISTENER.ORA in the sample directory. By default, this is the directory where Oracle SQL/Services expects to find the LISTENER.ORA file during the startup of OCI_DISP. The following example shows how to start the dispatcher:

```
SQLSRV_MANAGE> connect server;
SQLSRV_MANAGE> start dispatcher oci_disp;
SQLSRV_MANAGE> show dispatcher oci_disp;
```

The state of the dispatcher will change from starting to running. Once in a running state, the dispatcher is listening for OCI network traffic. In the rare case where OCI_DISP does not start, the default location for the OCI dispatcher log file is SYSSMANAGER:SQS*_OCI_DISP*.LOG where the asterisks depict the OpenVMS node name and version-specific naming convention for the process and log file. Examine this log file for errors.

If a version of SQL*Net for Rdb is already installed on the OpenVMS server prior to installing release 7.1.5, the Oracle SQL*Net management utilities are not used to start, stop, or reconfigure a SQL*Net for Rdb dispatcher. The SQL*Net for Rdb OCI_DISP dispatcher must be started, stopped, and configured using the SQLSRV_MANAGE utility, or the Oracle Rdb DBAPack client graphical user interface (GUI).

4.3.2 Configuring TNSNAMES.ORA

The sample file TNSNAMES.ORA_SAMPLE is provided to serve as a template for the mandatory client-side TNSNAMES.ORA file. If the client system currently uses a TNSNAMES.ORA file, this same file should be used for Oracle Rdb access as well. The Oracle Net8 Assistant GUI can be used to create and maintain this file. The TNSNAMES.ORA_SAMPLE file is a template provided for customers that do not already have Oracle SQL*Net applications installed and configured. If the client already has an existing TNSNAMES.ORA file, simply update that file to include:

1. A unique alias name for the Oracle Rdb connection parameters.

2. A SID or SERVICE keyword which contains the name of the Oracle SQL/Services OCI service name. Both keywords are interchangeable.
3. A valid HOST node name or TCPIP address that defines the OpenVMS system node name or TCPIP address.
4. The PORT parameter that contains the TCPIP port number for the OCI_DISP dispatcher (1527 by default).

If the client does not already have an existing TNSNAMES.ORA file, edit TNSNAMES.ORA_SAMPLE using any OpenVMS editor and follow the above instructions regarding keyword parameters. The resulting file from the edit session must be called TNSNAMES.ORA and must reside on the client system. The location for TNSNAMES.ORA on the client is platform specific. For Windows NT, Windows 95, Windows 98, and Windows 2000 the location is oracle_home/network/admin. Note that Oracle SQL*Net client software must also be installed on the client system before the client application can connect to Oracle Rdb on the server.

The following is an example of a TNSNAMES.ORA entry for an Oracle Rdb connection:

```
rdb_71 =
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (PORT=1527)
      (HOST=alpha1))
    (CONNECT_DATA=
      (SERVICE=oci_srv)
    ))
```

In the above example, RDB_71 is the user-defined alias name. This name must be unique within the TNSNAMES.ORA file. The PORT number must match the port number of the OCI dispatcher running on the server. The HOST parameter is either the OpenVMS node name or the TCPIP address of the server. Either keyword SID or SERVICE may be used to define the Oracle SQL/Services OCI service name. In this case the Oracle SQL/Services service name is OCI_SRV.

There are two classes of Oracle SQL/Services services; database and universal. A database class service includes the SQL ATTACH statement to a specific Oracle Rdb database. The previous example is a database class service definition. The universal class service does not include an SQL ATTACH statement and may be used to access different Oracle Rdb databases. When using universal services, an additional keyword, RDB_DATABASE, must be included in the TNSNAMES.ORA file. This

parameter contains the location of the Oracle Rdb database, including disk and directory names. The RDB_DATABASE keyword must immediately follow the SID or SERVICE definition.

The following is an example of a universal service TNSNAMES.ORA file:

```
rdb_u_71 =
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (PORT=1527)
      (HOST=alpha1))
    (CONNECT_DATA=
      (SERVICE=oci_712)
      (RDB_DATABASE=dka300:[my_dir]mf_personnel)
    ))
```

Using a universal service, it is possible to attach to multiple Oracle Rdb databases from a single connection. An ASCII script file on the OpenVMS server is required for multiple attaches. (You can also use a script to attach to a single database.) The SQL ATTACH statements are defined in the script file.

For example, an OpenVMS file called MULTI.SQL is created and contains the following lines:

```
-- MULTI.SQL
ATTACH 'FILENAME DKA300:[DIR]mf_personnel';
ATTACH 'ALIAS a FILENAME DKA500:[DIR1.DIR2]rdb_prod';
-- END OF FILE
```

In the TNSNAMES.ORA file, the RDB_DATABASE keyword is defined differently. An at sign (@) character is required before the name of the OpenVMS script file. The previous example used a database name for RDB_DATABASE but for multiple attaches, a filename is specified instead. For example:

```
rdb_2_db =
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (PORT=1527)
      (HOST=alpha1))
    (CONNECT_DATA=
      (SERVICE=oci_712)
      (RDB_DATABASE=@dka300:[my_dir]multi.sql)
    ))
```

Whenever the objects from the second database are accessed, the alias name (a in the following example) is required:

```
SELECT last_name FROM emp@a WHERE employee_id='00164';
```

Note that both of the universal examples reference the same universal service, OCI_712. This is possible because there is no SQL ATTACH statement associated with the OCI_712 service definition.

Before any connection can be established, the OCI dispatcher and the OCI service must be in running states on the server.

Users can then test their OCI applications (for example, SQL*Plus and Oracle Forms) by supplying the username, password, and TNS alias name during the connection.

4.3.3 Configuring Optional Files

The SQLNET.ORA_SAMPLE file is provided as a template in the event that the DBA wants to create a SQLNET.ORA file. The SQLNET.ORA file is not necessary for connecting to Oracle Rdb. It is an optional file that can be used for either specifying a default domain name other than WORLD or to enable SQL*Net logging.

The SQLNET.ORA file can reside on either the client or the server. If using SQLNET.ORA on the OpenVMS server, make sure that the AUTOMATIC_IPC parameter is set to OFF. For example:

```
AUTOMATIC_IPC = OFF
```

To specify a default domain name, other than WORLD, specify the following:

```
NAMES.DEFAULT_DOMAIN = <domain_name>
```

When tracing is enabled using the SQLNET.ORA file, the level of tracing is for the Oracle SQL*Net layer, not the SQL*Net for Rdb layer.

The valid parameters for enabling SQLNET.ORA tracing, depending on where the tracing occurs (client or server, or both) are:

- TRACE_LEVEL_CLIENT (default of OFF)
- TRACE_LEVEL_SERVER
- TRACE_FILE_CLIENT (default is SQLNET.TRC)
- TRACE_FILE_SERVER

- `TRACE_DIRECTORY_CLIENT` (operating system specific; must be a valid directory)
- `TRACE_DIRECTORY_SERVER`

For example:

```
TRACE_LEVEL_CLIENT = USER
TRACE_FILE_CLIENT = RDB.TRC
TRACE_DIRECTORY_CLIENT = C:\TEMP
```

4.4 Upgrading Oracle Rdb Release 7.0 Database to Oracle Rdb Release 7.1.0

If you want to upgrade your database from Oracle Rdb release 7.0 to Oracle Rdb release 7.1.0 and, if you ran the `RDB_NATCONN_DBMSOUTPUT_PREPARE.SQL` or the `RDB_NATCONN_DBMSAPPL_PREPARE.SQL` script on the Oracle Rdb 7.0 database, you must take the following steps:

1. Run `RDB_NATCONN_DBMSOUTPUT_DROP.SQL` script on the Oracle Rdb 7.0 database or on the converted Oracle Rdb release 7.1 database. You can either run this script on the Oracle Rdb 7.0 database and then convert to Oracle Rdb release 7.1 or convert from Oracle Rdb release 7.0 to Oracle Rdb release 7.1 and then run the `RDB_NATCONN_DBMSOUTPUT_DROP.SQL` script on the Oracle Rdb release 7.1 database.
2. Run the `RDB_NATCONN_UPGRADE71.SQL` script on the Oracle Rdb release 7.1 database.
3. Run the `RDB_NATCONN_DBMSOUTPUT_PREPARE.SQL` script on the converted Oracle Rdb release 7.1 database to install the `DBMS_OUTPUT` functionality.

4.5 Script Names

The following table details the script names that you should use with Oracle Rdb release 7.0 or Oracle Rdb release 7.1 databases on the OpenVMS platform.

Table 4–1 Script Names

For Use With Oracle Rdb Release 7.0 Databases	For Use With Oracle Rdb Release 7.1 Databases
RDB_NATCONN_PREPARE70.SQL	RDB_NATCONN_PREPARE71.SQL
RDB_NATCONN_DROP70.SQL	RDB_NATCONN_DROP71.SQL
RDB_NATCONN_UPGRADE70.SQL	RDB_NATCONN_UPGRADE71.SQL
RDB_NATCONN_DEV60_DROP70.SQL	RDB_NATCONN_DEV60_DROP71.SQL
RDB_NATCONN_DEV60_PREPARE70.SQL	RDB_NATCONN_DEV60_PREPARE71.SQL
RDB_NATCONN_DEV6I_DROP70.SQL	RDB_NATCONN_DEV6I_DROP71.SQL
RDB_NATCONN_DEV6I_PREPARE70.SQL	RDB_NATCONN_DEV6I_PREPARE71.SQL

The following scripts are available for either Oracle Rdb release 7.0 or Oracle Rdb release 7.1 databases on the OpenVMS platform:

- RDB_NATCONN_DBMSOUTPUT_PREPARE.SQL
- RDB_NATCONN_DBMSOUTPUT_DROPS.SQL
- RDB_NATCONN_DBMSAPPL_PREPARE.SQL
- RDB_NATCONN_DBMSAPPL_DROP.SQL

The *Guide to SQL*Net for Rdb7* documents these scripts without an embedded version number. Please note this difference when reading the *Guide to SQL*Net for Rdb7*.

4.6 New and Changed Features

This section highlights new and changed features for the SQL*Net for Rdb release 7.1.5 kit and the installation procedures.

4.6.1 Emulate Oracle V8.0.5.1

This SQL*Net for Rdb release 7.1.5 kit now emulates Oracle 8.0.5.1. It includes new features from the OCI8 protocol.

4.6.2 New Logical Names

Two new logical names have been added to this release of SQL*Net for Rdb:

- **SQLNET_RECO_USER**

If you are using SQL*Net for Rdb from a database link and you are doing two-phase commit, if either the Oracle system or the Oracle Rdb system fail while in the middle of a transaction, the Oracle Transaction Manager tries to reconnect to the Oracle Rdb database to verify the results of the transaction. In order for SQL*Net for Rdb to be able to make this connection, it needs a username and password. The user only needs access to the ORA_COMM_TRANS table which records the results of transactions as they are in progress.

- **SQLNET_BLOB**

In previous versions of SQL*Net for Rdb, Oracle Rdb segmented strings were treated as long character strings by the Oracle tool. For this release, much of the OCI LOB interface is implemented, so it is possible to treat segmented strings as Oracle LOBs. To enable this, you must define this logical to be "Y" or "y". If you define the logical as anything else, or if you do not define it, segmented strings in Oracle Rdb will be treated like long character strings as they have been in previous releases.

4.6.3 New Command File RDB_NATCONN.COM

There is a new command file called RDB_NATCONN.COM. For this release, it can be used to add users and their encrypted passwords to an Oracle Rdb database. Most Oracle tools require that users and their passwords be in the database to allow the tools to connect. This command file allows a DBA, or someone with access to the database, to add the users.

There are two parameters associated with this command file:

- The operation you would like to do
- The database to which you want to attach

If you do not specify the parameters, the command file prompts for them. If you specify adduser as the operation, you are prompted for usernames and passwords. Enter a blank name, by pressing the Enter key, to indicate you have entered all usernames and passwords needed. The command file adds them all to the database. For example:

```
@rdb_natconn adduser [my_dir]my_database
Username: username1
Password: password1
Username: username2
Password: password2
Username: <Enter>
```

Note: In a future release, this command file will also be used to prepare, upgrade, and drop the Oracle dictionary in the Oracle Rdb database. This functionality is not available in this release.

4.6.4 Displaying Oracle Messages

In order to display Oracle messages while using languages other than AMERICAN, add the following commands to the process initialization file for your SQL/Services service:

```
$ DEFINE ORA_NLS33 SYS$COMMON:[SQLSRV71.SQLNET.NETWORK]
$ DEFINE ORA_NLS SYS$COMMON:[SQLSRV71.SQLNET.NETWORK]
```

4.6.5 Support for Hot Standby

SQL*Net for Rdb now supports Hot Standby databases.

4.6.6 SQL*Net for Rdb Product

SQL*Net for Rdb supports Oracle Rdb release 7.0 and release 7.1 databases. Earlier releases of Oracle Rdb databases (such as release 6.1) are not supported.

You might see SQL*Net for Rdb identified as "Oracle Rdb OCI Server" in some places such as the welcome banner and in some online documentation. Because SQL*Net for Rdb serves OCI (Oracle Call Interface) clients and clients that use OCI technology to access a database, the software sometimes makes references to OCI in its documentation and its setup procedures.

4.6.7 Compatibility with Oracle Forms

If you use Developer/2000 prior to release 2.0 or prior to Oracle Forms release 5.0, you must set the following properties in your Oracle Forms module for compatibility with SQL*Net for Rdb:

- Set the Module property "Savepoint Mode" to FALSE
- Set the Data Block property "Locking Mode" to DELAYED

You can modify the properties using the Property Palette or use the Property Class RDB_PROPERTY_CLASS of the file RDB_PROP.FMB provided in the Oracle SQL/Services examples directory on your server.

Beginning with Developer/2000 release 2.0 or Oracle Forms Release 5.0, these properties will be handled automatically during run time by Developer/2000.

4.7 Known Problems and Restrictions

This section highlights problems and restrictions relating to SQL*Net for Rdb and included workarounds where appropriate.

4.7.1 Restrictions and Limitations

The restrictions and limitations in [Table 4–2](#) apply to this release of SQL*Net for Rdb:

Table 4–2 SQL*Net for Rdb Restrictions and Limitations

Category	Description
Character sets	The Oracle US7ASCII and DEC_MCS character sets are supported by default. See the <i>Guide to SQL*Net for Rdb7</i> for information about using other character sets.
Oracle Rdb interval data type	This data type cannot be mapped to an Oracle data type. When dealing with this Oracle Rdb data type, you must use the CAST function to convert the value to an appropriate Oracle data type.
Dates	Oracle dates can pre-date the OpenVMS 17-NOV-1858 date. Such dates are not supported by Oracle Rdb or SQL*Net for Rdb.
Transaction reusable services	Oracle SQL/Services transaction reusable services are not supported. Only session reusable services are supported.
Database access by service owner	Oracle SQL/Services services with database access by service owners are not supported. Only services with database access by connect user are supported.
Grant use on OCI services	Grant use on OCI services is not supported. All users have access to OCI services. However, the database is still protected because access to the database must be through connect (client) username.
Multischema databases	SQL*Net for Rdb does not support multischema Oracle Rdb databases. However, SQL*Net for Rdb adds an emulation layer that provides a multischema environment that is similar to what you get with Oracle.
Error mapping is not exact	The Oracle error codes do not always map well to Oracle Rdb message vectors. As a result, SQL*Net for Rdb might issue error code 9100. Error code 9100 is a generic server error code that does not convey any information about the error. The text accompanying the message provides more information about handling this error.
SQL dialect	SQL*Net for Rdb sets the Oracle_Level1 dialect and requires the dialect to perform its tasks. Changing the dialect may result in unexpected SQL*Net for Rdb failures.

Table 4–2 (Cont.) SQL*Net for Rdb Restrictions and Limitations

Category	Description
Multisession server is not supported	The Oracle server can support multiple sessions from multiple client applications in one server process. SQL*Net for Rdb cannot because of the locking differences between the two database products.
Cursors	The maximum number of open cursors is currently 998.
ROWID support	ROWID support exists only if there are less than 2.1 billion pages in a logical area. Logical areas greater than 2.1 billion pages are not likely to occur unless you implement VLDB with a single-file database.
Piecewise inserts	Piecewise inserts (specific to Oracle SQL) are not supported in this release.
Oracle Rdb does not differentiate between RAW (binary) and VARCHAR (text) data	If you use the VARCHAR data type in place of the RAW data type, your application works with one exception. SQL*Net for Rdb does not convert the binary data to text data if you select the data using the VARCHAR2 data type. You must explicitly convert the data using the HEXTORAW or RAWTOHEX SQL function. Explicitly converting the data works with both Oracle Rdb and Oracle.
Determining a NOT NULL constraint through a describe call	If the statement SELECT col1 FROM tab1 were parsed and described in OCI, the Oracle server returns a message that col1 either does or does not have a NOT NULL constraint. SQL*Net for Rdb does not return this information in the describe call. You can, however, obtain this information from the metadata tables. This restriction causes applications like SQL*Plus to always describe a column as not having a NOT NULL constraint when it in fact may have a NOT NULL constraint.

4.7.2 SQL Functions Return Incorrect User Information

The SQL functions USER, CURRENT_USER, SESSION_USER, and SYSTEM_USER do not return the correct results to an OCI client connected to a SQL*Net for Rdb release 7.1.5 (OCI) service that is using an Oracle Rdb release 7.0 database. These functions return the username of the user who had connected to the service prior to the current connection.

For example, for this scenario assume there is a universal service named AAA. Also assume that a user named Smith has already connected to an Oracle Rdb database through service AAA. If a user named Johnson connects to an Oracle Rdb database through service AAA after Smith's connection and issues a SELECT USER command, the USER function will return information about Smith. For example:

```
$ SQLPLUS SMITH/password@AAA
$ SQLPLUS JOHNSON/password@AAA
SQL> SELECT USER FROM DUAL;
SMITH
SQL>
```

As a workaround, Johnson can disconnect and immediately reconnect to the database through service AAA to get the correct user information when issuing the `SELECT USER` command. For example:

```
SQL> EXIT
$ SQLPLUS JOHNSON/password@AAA
SQL> SELECT USER FROM DUAL;
JOHNSON
SQL>
```

Services that use an Oracle Rdb release 7.1 database return the correct results without using this workaround.

4.7.3 Alterations to Dispatcher Network Port Resets Protocol to SQLSRV

The Oracle SQL/Services dispatcher used to serve Oracle clients must specify an OCI protocol. The installation procedure creates a default dispatcher, `OCI_DISP`, that you can use to serve OCI clients, or you can optionally create your own OCI dispatcher.

If you use the `SQLSRV_MANAGE` command line utility or the Oracle SQL/Services Manager GUI to create a dispatcher or to change any network port attributes, you must specify the OCI protocol on the command line. For example, if you alter an existing dispatcher and modify the `NETWORK PORT`, you also must specify `PROTOCOL OCI`. Otherwise, the protocol changes from OCI back to SQL/Services. The following example shows a command line that specifies an OCI protocol:

Example 4-1 Specifying the OCI Protocol with SQLSRV_MANAGE71

```
$ MCR SQLSRV_MANAGE71
SQLSRV> alter disp oci_disp
_SQLSRV> network_port sqlnet
_SQLSRV> listener "oci_listener"
_SQLSRV> protocol oci;
SQLSRV>
```

If you try to access an Oracle SQL/Services dispatcher defined with the SQL/Services protocol from an OCI client, the connect attempt fails and the following errors are logged in the dispatcher log file:

Example 4-2 Errors Logged to the Dispatcher Log File

```
-----EVENT BEGIN:  EVENT_LOG at Tue Feb 11 1997 16:15:42.456-----
%SQLSRV-I-EVENT_LOG, event logged at line 838 in file SRVUTL.C;1
```

```

%SQLSRV-F-INTERR, Internal error -2007 in Oracle SQL/Services dispatcher
    version V7.1-5 at line 1403 in module SRVPRSMS
%SQLSRV-E-ERROR_TEXT, Error text: invalid packet ID tag 1 or data type 4
    message data 02030401 flag 0
-----EVENT END   :   EVENT_LOG at Tue Feb 11 1997 16:15:42.502-----

-----EVENT BEGIN:   EVENT_LOG at Tue Feb 11 1997 16:15:42.529-----
%SQLSRV-I-EVENT_LOG, event logged at line 1062 in file MSG_COM_SQS.C:1
%SQLSRV-E-SQSMGERROR, Oracle SQL/Services MSG-layer error, client: unknown
%SQLSRV-E-SQSBADPKTHDR, Bad Oracle SQL/Services packet header
-----EVENT END   :   EVENT_LOG at Tue Feb 11 1997 16:15:42.544-----

-----EVENT BEGIN:   EVENT_LOG at Tue Feb 11 1997 16:15:42.572-----
%SQLSRV-I-EVENT_LOG, event logged at line 284 in file DISP.C:1
%SQLSRV-W-EXCEPTION_RAISE, Exception raised: DBS_CANCEL_CONNECT
%SQLSRV-I-CONNECTNAME, Connect : CONNECT_0000001
%SQLSRV-I-CONNECTSTATE, Connect state: 1
%SQLSRV-I-NODENAME, Node : unknown
-----EVENT END   :   EVENT_LOG at Tue Feb 11 1997 16:15:42.573-----

```

To correct this problem, use the `SQLSRV_MANAGE` command line utility or the Oracle SQL/Services Manager GUI to:

1. Alter your dispatcher and specify `PROTOCOL OCI`.
2. Stop and restart your Oracle SQL/Services dispatcher so the change can take effect.

4.7.4 Problem With RMU Dispatcher

You cannot configure the Oracle RMU dispatcher for Oracle SQL/Services with alternate network ports. Therefore, the RMU dispatcher for Oracle SQL/Services release 7.0 uses the same network ports as the RMU dispatcher for Oracle SQL/Services release 7.1.5.

For this reason, you cannot have an RMU dispatcher running for more than one release of Oracle SQL/Services. If the Oracle SQL/Services release 7.1.5 multiversion installation detects an existing version of Oracle SQL/Services release 7.0, the installation procedure creates the RMU dispatcher with the "autostart off" attribute in the release 7.1.5 configuration. If an existing release 7.1.5 configuration is found, the installation procedure will not change the RMU dispatcher, so you should alter the RMU dispatcher to disable autostart.

4.7.5 SQL*Plus Release 3.3: Error Accessing Package DBMS_APPLICATION_INFO

If you are using SQL*Plus release 3.3, the following messages are returned if the client expects a version of Oracle7 that has the "Procedural option" (server-side PL/SQL) installed:

Example 4-3 Errors Returned When Accessing DBMS_APPLICATION_INFO

ORA-00904: invalid column name

%SQL-F-RTNNOTDEF, function or procedure ORA_OUTPUT_DISABLE is not defined

Error accessing package DBMS_APPLICATION_INFO.

Error accessing package DBMS_APPLICATION_INFO
You may need to install the Oracle Procedural option
SET APPINFO requires Oracle Server Release 7.2 or later
Use of Oracle SQL feature not in SQL92 Entry Level.

Connected to:
Oracle Rdb OCI Server Release 1.0.2.0.0 - Beta, Level 1.7
Oracle Rdb SQL Release 7.0.1.1.0 - Production

SQL>

The ORA-00904 error is returned because SQL*Net for Rdb does not support server-side PL/SQL for this release.

You can ignore the error. The procedure is attempting to access user profile information that is not necessary for the correct functioning of this application.

To eliminate the ORA-00904 error, install the DBMS_OUTPUT package by running the RDB_NATCONN_DBMSOUTPUT_PREPARE.SQL script. You can run the RDB_NATCONN_DBMSAPPL_PREPARE.SQL script to eliminate the second set of messages.

4.7.6 Oracle Reports: Problem with Rdb Accepting a Built-In Function

If you are using Oracle Reports release 2.5, a bugcheck dump might occur when you click on the TABLE/COLUMNS button in the DATA MODEL QUERIES window.

The problem occurs when Rdb SQL does not accept a built-in function in the ORDER BY clause.

To fix the problem, define a stored function that meets the exact needs of that query. The DECODE built-in function actually takes a varying number of parameters and

this stored function does not. The following workaround is tailored specifically for this situation. Use the SQL interactive language to execute the following:

Example 4–4 Stored Function Module Workaround

```
CREATE MODULE workaround_rpt1 LANGUAGE SQL
-- Work around the fact that the ORDER BY clause doesn't accept
-- built-in functions, but does accept stored functions.
-- Needed for Reports V2.5; not needed for Reports V3.0
FUNCTION "DECODE"(
    IN :one   VARCHAR(2000),
    IN :two   VARCHAR(2000),
    IN :three VARCHAR(2000),
    IN :four  VARCHAR(2000))
    RETURN VARCHAR(2000)
    COMMENT IS 'Four parameter DECODE work around';
BEGIN
RETURN DECODE(:one, :two, :three, :four);
END;
END MODULE;
GRANT EXECUTE ON MODULE workaround_rpt1 TO PUBLIC;
```

The preceding stored module is a workaround to this particular problem.

4.7.7 OCI: Error When Attempting to Fetch a Binary ROWID

The OCI FETCH routine returns the ORA-03106 error when you attempt to fetch a binary ROWID (data type DTYRID) with an OpenVMS Alpha server. The FETCH routine does not work currently on the OpenVMS Alpha server.

To work around this problem, fetch a ROWID in text. This method works well for all servers, including OpenVMS Alpha servers.

4.7.8 OCI: Error When Program Linked in Nondeferrable Mode

The OCI FETCH routine returns the ORA-03106 error when your program has been linked in nondeferrable mode. This is an uncommon link option that is not available on Microsoft Windows NT and some other platforms.

However, if clients running on a Compaq Computer Corporation OpenVMS system try to link debug images in the nondeferrable mode, it will fail. This mode does not currently work on OpenVMS systems.

4.7.9 Stored Procedures with Bind Variables

A stored procedure with bind variables can be called only once per SQL parse. In other words, a stored procedure must be parsed before each execution. Failure to do so may result in an "ORA-03106 fatal two-task communication protocol error".

4.8 Software Errors Fixed

This section highlights software errors fixed relating to SQL*Net for Rdb release 7.1.5.

4.8.1 Environment Problems Reported in the OCI Dispatcher Log File Fixed

If you encounter the following error in the OCI dispatcher log file when performing an operation such as starting the Oracle SQL/Services OCI dispatcher or accessing an Oracle Rdb database from a Java client through JDBC, there is most likely a problem with your Oracle SQL/Services environment:

```
%SQLSRV-I-EVENT_LOG, event logged at line 1082 in file COM_TNS.C:1
%SQLSRV-E-ERROR_TEXT, Error text: This version of SQL*Net does NOT have
non-blocking send support
```

The following are examples of environment problems that can cause this error:

- The LISTENER.ORA file does not exist or is not in the required location. If Oracle is installed, this file should be in the ORA_ROOT:[NETWORK.ADMIN] directory. Otherwise, it should be in the SYSSCOMMON:[SQLSRV71.SQLNET.NETWORK.ADMIN] directory. If Oracle SQL/Services is installed on a system where Oracle is installed, and the installer takes the default of SYSSCOMMON:[SQLSRV71.SQLNET.NETWORK.ADMIN], rather than entering the Oracle root, or enters an invalid Oracle root, the environment will be in a state where LISTENER.ORA is expected in two locations. In order to solve this problem, Oracle SQL/Services needs to be reinstalled and you need to supply the correct Oracle root directory.
- LISTENER.ORA specifies an invalid protocol for the system. For instance, if LISTENER.ORA contains the following information and IPC is not a valid protocol on the system, it must be removed.

The following example is invalid:

```
OCI_LISTENER =
  (ADDRESS_LIST =
```

```

    (ADDRESS =
      (PROTOCOL = IPC)
      (KEY = mysys.world)
    )
  (ADDRESS =
    (PROTOCOL = IPC)
    (KEY = mysys)
  )
  (ADDRESS =
    (COMMUNITY = tcp_com.world)
    (PROTOCOL = TCP)
    (HOST = mysys)
    (PORT = 1526)
  )
)

```

The following example is valid:

```

OCI_LISTENER =
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = tcp_com.world)
      (PROTOCOL = TPC)
      (HOST = mysys)
      (PORT = 1526)
    )
  )
)

```

- Your database must have a username and password defined for the user logging into the database. If a username and password does not exist, use the SYSSLIBRARY:RDB_NATCONN.COM. See [Section 4.6.3](#) for additional details.

4.8.2 Access Violations Fixed

The following access violation bugs have been fixed in this release of SQL*Net for Rdb:

- In previous releases when you attempted to connect from Oracle SQL*Net, you received an ORA-00022 error message (invalid session id; access denied). This error no longer occurs and you can now connect from Oracle SQL*Net.
- The access violation that occurred when the special recovery user logged on has been fixed.

4.8.3 Failure While Holding a Mutex During Executor Cleanup

If a client connection terminated abnormally while executing in an executor process, the executor process is cleaned up and shutdown. In previous releases of Oracle SQL/Services, this executor cleanup sometimes caused an executor to shutdown while holding a mutex.

This problem is fixed in this release.

4.8.4 Using Bind Variable Twice in a Query

If you used a bind variable twice in a query, you might have seen the error ORA-01006. This has been fixed for this release.

4.8.5 ANSI_DATE and SYSDATE Functions Work Correctly

In previous releases, the use of ANSI_DATE and SYSDATE functions failed unless you used them in conjunction with CAST or the TO_DATE, or TO_CHAR functions.

The following SQL statements now work:

- `SELECT * FROM TEST_DATE WHERE ANSI_DATE = '01-jan-1999';`
- `SELECT * FROM TEST_DATE WHERE ANSI_DATE = SYSDATE;`
- `UPDATE TEST_DATE SET ANSI_DATE = SYSDATE;`
- `INSERT INTO TEST_DATE (ANSI_DATE) VALUES (SYSDATE);`
- `DELETE FROM TEST_DATE WHERE ANSI_DATE = SYSDATE;`
- `UPDATE TEST_DATE SET VMS_DATE = TO_DATE(SYSDATE);`
- `SELECT * FROM TEST_DATE WHERE ANSI_DATE = TO_DATE('01-JAN-1999') - 1;`

4.8.6 Wrong Error Returned When Inserting a Duplicate Value into a Unique Index

In previous releases, an incorrect error message was being returned when you attempted to insert a duplicate value into a unique index. The incorrect error message returned was:

```
ORA-09100: Target system returned following message
```

The ORA-09100 error message was followed by the RDB_E_NO_DUP error message.

The following error message is now returned which is the correct Oracle error:

```
ORA-00001, unique constraint (<table_name>.<index_name>) violated
```

4.8.7 SYSDATE Function is Available from a Dblink Connection

The SYSDATE function is available from a dblink connection. When used in this way, SYSDATE is evaluated by the Oracle Rdb Server, which is not standard Oracle semantics. Please be aware of this difference and certain that you want to use SYSDATE in this way.

4.8.8 Embedded SQL Program Using SET TRANSACTION Works Correctly

In previous releases, an embedded SQL program that performed a SET TRANSACTION statement returned an SQL_F_BAD_TXN_STATE error message.

This has been fixed in this release.

4.8.9 ALL_TAB_COLUMNS and USER_TAB_COLUMNS Views Return Correct Scale and Precision

The ALL_TAB_COLUMNS and USER_TAB_COLUMNS views created from the RDB_NATCONN_PREPARE.SQL script now return the correct scale and precision for smallints and integers with decimal points.

4.9 Documentation Changes

This section provides descriptions of product enhancements, changes, and corrections for the SQL*Net for Rdb documentation set. This material was documented in the release notes of previous releases and will be merged into the permanent documentation the next time it is updated.

4.9.1 Guide to SQL*Net for Rdb

This section provides information about product enhancements, changes, and corrections that will be merged into the *Guide to SQL*Net for Rdb* the next time it is updated.

4.9.1.1 Changes to Database Privileges

Changes have been made to the RDB_NATCONN_PREPARE.SQL and RDB_NATCONN_UPGRADE.SQL scripts to allow for a "secure" database where users

only need SELECT privileges on the database. SELECT is the only privilege granted to PUBLIC. SQLNET4RDB is granted the required privileges for tables that require UPDATE, INSERT, or DELETE privileges.

The required rights identifier, SQLNET4RDB, will be created as part of the SQL/Services installation. For new installations or for new databases that have not been previously prepared for SQL*Net for Rdb, the RDB_NATCONN_PREPARE71.SQL script must be run on each database that is to be accessed by SQL*Net for Rdb. For databases that have been previously prepared for SQL*Net for Rdb, the RDB_NATCONN_UPGRADE71.SQL script must be run. This script will upgrade all tables, views, and modules to the latest release and will change the privileges on existing tables and views so only SQLNET4RDB has UPDATE, INSERT, or DELETE privileges.

4.9.1.2 New Scripts to Mimic Oracle Public Packages

RDB_NATCONN_DBMSOUTPUT_[PREPARE | DROP] and RDB_NATCONN_DBMSAPPL_[PREPARE | DROP] are two new scripts included with this release of SQL*Net for Rdb. These scripts may optionally be run on databases that are prepared for SQL*Net for Rdb.

The RDB_NATCONN_DBMSOUTPUT_PREPARE script implements the functionality of the Oracle public package DBMS_OUTPUT, which allows outputting messages from stored procedures, packages, and triggers.

The RDB_NATCONN_DBMSAPPL_PREPARE script implements the functionality of the Oracle public package DBMS_APPLICATION_INFO, which allows users to register an application name with the database for auditing or performance tracking.

If you see error messages that indicate that functionality from one or both of these packages is not available when starting your client, you can run these scripts to make the functionality available.

RDB_NATCONN_DBMSOUTPUT_DROP and RDB_NATCONN_DBMSAPPL_DROP scripts will drop just the tables, views, and modules that implement that package's functionality.

4.9.1.3 Encrypted Passwords

The new scripts, RDB_NATCONN_PREPARE.SQL and RDB_NATCONN_UPGRADE.SQL define two new procedures that implement secure login where the password is encrypted and stored on the server. The client and server send only encrypted passwords at login time. These procedures must be invoked through

SQL*Plus or a similar client tool, or by the new command file RDB_NATCONN.COM.

The procedure `ORA_SET_PASSWORD` accepts the user name and the user's password as input. The password is encrypted and stored in the table `USER$` in the current Oracle Rdb database. This encrypted password is retrieved by SQL*Net for Rdb when the user logs on and is sent to the client for validation.

If the user does not exist in the `USER$` table, the user name and encrypted password will be inserted. If the user already exists in the `USER$` table, the new password is encrypted and replaces the old password in the table. Users may be removed by invoking the `ORA_DROP_PASSWORD` procedure with just the user name as input.

Some client applications require this functionality for users to be able to successfully log on. Other clients may try to log on with the encrypted password but will allow for the non-encrypted login if the encrypted functionality is not available. Some older clients or older versions of clients do not use this new functionality. The SQL*Net for Rdb log file shows if the client will use the new functionality. The client is looking for the encrypted password if the following error is included in the log file right after the Oracle Rdb SQL Release identification line:

```
Oracle error: 1010 for function: O3LOGA
Function: 0 2 LOGON
```

4.9.1.4 Correction to Character Set for DEC-Kanji

Table 3-3 in the *Guide to SQL*Net for Rdb* lists the character sets supported by SQL*Net for Rdb. The character set listed for DEC-Kanji is incorrect. The correct character set for DEC-Kanji is JA16EUC.

The *Guide to SQL*Net for Rdb* will be corrected in a future release.

