

Oracle® Rdb

Oracle SQL/Services Release 7.2 Release Notes

January 2006

This document contains release note information specific to Oracle SQL/Services, release 7.2 for OpenVMS Alpha and HP OpenVMS Industry Standard 64 for Integrity Servers operating systems. Also included in this document are release notes pertaining to OCI Services for Oracle Rdb release 7.2.

Oracle SQL/Services Release 7.2 Release Notes

Copyright © 2006, 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:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth 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, Oracle Rdb7, and SQL*Net are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Send Us Your Comments	xiii
Preface	xv
Intended Audience	xv
Operating System Information	xvi
Structure	xvi
Related Manuals	xvi
Conventions	xvii
1 Oracle SQL/Services: New and Changed Features	
1.1 Installing Oracle SQL/Services Release 7.2	1-1
1.1.1 Software Installation Requirement and Compatibility Information	1-2
1.2 Running the Oracle SQL/Services Configuration File Utility	1-5
1.2.1 Converting Previous Configuration Files to Release 7.2	1-5
1.2.2 Converting Previous Configuration Files to Release 7.0	1-6
1.3 Documentation	1-7
1.4 Summary of Oracle SQL/Services New and Changed Features for Release 7.2	1-7
1.4.1 Oracle SQL/Services Available on OpenVMS I64	1-8
1.5 Obsolete Features	1-8
2 Oracle SQL/Services: Software Errors Fixed	
2.1 Oracle SQL/Services Errors Fixed in Release 7.2	2-1

2.1.1	Monitor Aborts When Connection Cancelled.....	2-1
-------	---	-----

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 7.2 Server Known Problems and Restrictions.....	3-1
3.2.1	Support for OpenVMS VAX and Standard Kits.....	3-1
3.2.2	Concealed Attributes are Required for Rooted Directory Logicals	3-2
3.2.3	Do Not Kill Oracle SQL/Services Processes	3-3
3.2.4	Do Not Shut Down or Restart the SQLSRV_MANAGE System Management Service..	3-3
3.2.5	Do Not Abbreviate AUTOCONNECT or AUTOSTART Keywords.....	3-3
3.2.6	Oracle SQL/Services Management Utilities Do Not Prevent Multiple Dispatchers With the Same Port IDs	3-4
3.2.7	Database Service Attached to a Remote Database Does Not Know if the Database is Closed.	3-4
3.2.8	Log and Bugcheck Dump File Path Server Attributes are Ignored on OpenVMS Systems.....	3-5
3.2.9	Process Startup Fails Due to Unhandled Errors in Systemwide OpenVMS Login Procedure ..	3-6
3.2.10	DBSRC_FILE Service Attribute is Not Supported on OpenVMS	3-6
3.2.11	Installation Clarification: How to Determine Ports to Which to Direct Clients	3-6
3.2.12	How to Re-create Your RMU_DISP Dispatcher and RMU_SERVICE Service.....	3-8
3.3	Oracle SQL/Services Release 7.2 Client Known Problems and Restrictions	3-11
3.3.1	Oracle SQL/Services OpenVMS Client is Now Compiled With HP C	3-11
3.3.2	Use a Jacket Header File When Calling the Oracle SQL/Services API From C++	3-11
3.3.3	Client Enters Nonrecoverable Error State if a Statement With No Parameter Markers is Executed Using Batched Execution	3-12
3.3.4	Incorrect Error Message is Returned if a Client Cancels Batched Execution.....	3-12
3.3.5	PATHWORKS for DOS and Windows 95	3-12
3.3.6	Client Disconnect Does Not Abort Running Transaction For Transaction Reusable Services .	3-13
3.3.7	Applications Using Clients Shipped With Oracle SQL/Services V3.1 or Earlier Are Not Supported, But Are Not Rejected	3-13
3.3.8	Repeat Count on SQLSRV_FETCH_MANY Must be Less Than or Equal to 65535 ...	3-13
3.3.9	SQSAPIW.INI and SQSAPI32.INI Example is Misleading.....	3-13
3.4	Other Oracle SQL/Services Server Known Problems and Restrictions.....	3-14
3.5	Other Oracle SQL/Services Client Known Problems and Restrictions.....	3-16

3.6	Oracle SQL/Services Documentation Errors or Omissions	3-17
-----	---	------

4 OCI Services for Oracle Rdb: Release Notes

4.1	Software Requirements	4-1
4.2	Installing OCI Services for Oracle Rdb	4-1
4.2.1	Problem Reporting	4-1
4.3	Upgrading Oracle Rdb Release 7.0 Database to Oracle Rdb Release 7.1 4-2	
4.4	New and Changed Features for OCI Services for Oracle Rdb Release 7.2	4-2
4.4.1	OCI Services for Oracle Rdb Available on OpenVMS I64	4-2
4.5	Known Problems and Restrictions	4-2
4.5.1	Support for OpenVMS VAX and Standard Kits	4-2
4.5.2	Restrictions and Limitations	4-2
4.5.3	Alterations to Dispatcher Network Port Resets Protocol to SQLSRV	4-4
4.5.4	Problem With RMU Dispatcher	4-5
4.5.5	SQL*Plus Release 3.3: Error Accessing Package DBMS_APPLICATION_INFO	4-6
4.5.6	Oracle Reports: Problem with release 7.2 Accepting a Built-In Function	4-6
4.5.7	OCI: Error When Attempting to Fetch a Binary ROWID	4-7
4.5.8	OCI: Error When Program Linked in Nondeferrable Mode	4-7
4.5.9	Stored Procedures with Bind Variables	4-7

5 Previous Releases: New Features and Fixed Problems

5.1	New and Changed General Features in Previous Releases of Oracle SQL/Services	1-1
5.1.1	New Logical SQLSRV\$MAX_EXECUTOR_FAILURES	1-1
5.1.2	Logical SQLSRV\$LOG_CONNECTIONS	1-2
5.1.3	Oracle Server Two-Phase Commit Support	1-2
5.1.4	Linux Client Available	1-3
5.1.5	Executor Failure Maximum Logical	1-3
5.1.6	Enhancements to SQLSRV\$DEINSTALL_DELETE Procedure	1-4
5.1.7	New Logical Name	1-4
5.1.8	Version Check	1-5
5.1.9	New and Changed SQLSRV_MANAGE Features	1-5
5.1.9.1	Recall function	1-5
5.1.9.2	SQLSRV_DISP_LOGPATH	1-5
5.1.9.3	SQLSRV_DISP_DUMP_PATH	1-5

5.1.9.4	Extract Command	1-5
5.1.9.5	ALTER DISPATCHER and CREATE DISPATCHER	1-7
5.1.9.6	ALTER DISPATCHER and CREATE DISPATCHER	1-8
5.1.9.7	ALTER SERVICE and CREATE SERVICE	1-8
5.1.9.8	GRANT USE ON SERVICE	1-8
5.1.9.9	REVOKE USE ON SERVICE	1-8
5.1.9.10	SHOW	1-9
5.1.10	New and Changed Client Features	1-9
5.1.11	New and Changed Server Management Features	1-9
5.2	Oracle SQL/Services Errors Fixed in Prior Releases	1-9
5.2.1	Shared Memory Not Released with Continuous Start and Shutdown of Server	1-9
5.2.2	Erroneous RDB\$_NO_PRIV Errors Using OCI Services	1-10
5.2.3	OCI Universal Services Not Correctly Impersonating the Connect User	1-10
5.2.4	Long Queries Hang	1-10
5.2.5	Poor Performance From OCI Queries	1-10
5.2.6	.Monitor Crashes with "bind mon: unexpected monitor state"	1-11
5.2.7	Increased ENQLM Quota	1-11
5.2.8	Severity of Error Changed	1-11
5.2.9	SERVICE_NAME Now Allowed in Tnsnames.ora	1-12
5.2.10	Installation Procedure Overwrote Configuration File	1-12
5.2.11	Preattached Database Service with Default_connect_username Could Bugcheck	1-13
5.2.12	Impersonation Is Supported	1-13
5.2.13	Logicals Now Defined for ORA_NLS, ORA_NLS32, and ORA_NLS33	1-13
5.2.14	Oracle SQL/Services Management Utility Caused a Bugcheck to Occur	1-13
5.2.15	Oracle SQL/Services Dispatcher Failed to Completely Shut Down When There Were Clients Connected 1-13	
5.2.16	Problems Running Oracle SQL/Services V7.0 Multiversion with a Higher Version of Oracle SQL/Services Standard 1-14	
5.2.17	Oracle SQL/Services Executor Incorrectly Returned Multi-Octet Character Set Data from Columns of Type NATIONAL CHARACTER 1-14	
5.2.18	Multiversion Installations Erroneously Specified a Version Number	1-14
5.2.19	Executors Failed During Startup with RMS-E-PRV Errors	1-14
5.2.20	Problem Fetching More Than 65535 Rows in fetch_many Mode	1-15
5.2.21	Monitor Process Termination Left Dangling Oracle SQL/Services Processes	1-15
5.2.22	All Processes Did Not Explicitly Flush Log File Output	1-15
5.2.23	Installation Did Not Allow DECnet Object Name to Begin with a Numeric Value	1-16

5.2.24	Oracle SQL/Services Protocol Message to Monitor Caused a Bugcheck to Occur.....	1-16
5.2.25	Failure While Holding a Mutex During Executor Cleanup	1-16
5.2.26	Upgrade of Oracle SQLNET Libraries	1-16
5.2.27	Memory Leak of Monitor Process	1-17
5.2.28	Executing External Routines from Universal OCI Services	1-17
5.2.29	Support for Multiline Statements in SQL Initialization File.....	1-17
5.3	New and Changed Features for Previous OCI Services for Oracle Rdb Releases.....	1-17
5.3.1	Date-Time Data Type Enhancements	1-18
5.3.2	ALTER SESSION Enhancements	1-18
5.3.3	Emulate Oracle release 9.2.0.4.....	1-18
5.3.4	Dictionary Prepare/Upgrade/Drop Program Replaces Scripts.....	1-19
5.3.5	More Efficient Dictionary Queries	1-19
5.3.6	Hidden Objects in OCI Services for Oracle Rdb	1-19
5.3.7	New Tables in Oracle Data Dictionary	1-19
5.3.8	Enhanced RDB_NATCONNnn.COM File.....	1-20
5.3.9	Optional validation of connecting program name.....	1-20
5.3.10	New Users Visible Using OEM Console.....	1-20
5.3.11	New Error Message for Unregistered Usernames.....	1-21
5.3.12	Thin JDBC Access to Oracle Rdb Databases	1-21
5.3.13	New Command File RDB_NATCONN.COM	1-21
5.3.14	New Logical Names.....	1-22
5.3.15	Support for Debug Flags	1-22
5.3.16	Support for Hot Standby	1-24
5.3.17	SQL*Net for Rdb Product.....	1-24
5.3.18	Compatibility with Oracle Forms	1-24
5.3.19	Support for New ALTER SESSION SET CONSTRAINTS Syntax	1-25
5.3.20	Support for RDB\$NATCONN_CUPPnn Utility	1-25
5.4	Software Errors Fixed in Previous OCI Services for Oracle Rdb Releases.....	1-25
5.4.1	Invalid ROWID Messages	1-25
5.4.2	Unable to Connect to Multiple Databases Using a Universal Service.....	1-25
5.4.3	Milliseconds Supported for TIMESTAMP and INTERVAL Data Types.....	1-26
5.4.4	Error ORA-03106 Returned Instead of Expected ORA-01722 Error.....	1-26
5.4.5	Additional Error Messages in FORMS	1-26
5.4.6	TIMESTAMP Data Type Caused Error.....	1-26
5.4.7	Error Using DESCRIBE Command.....	1-26

5.4.8	Error Storing Null Value	1-26
5.4.9	DOUBLE PRECISION Column Converted Incorrectly	1-27
5.4.10	JDBC Errors	1-27
5.4.11	Incorrect Oracle Errors Returned to OCI Clients	1-27
5.4.12	Rdb Errors Unintentionally Suppressed	1-27
5.4.13	RDB_NATCONNnn Command Procedure No Longer Turns On Verify	1-28
5.4.14	Changes to RDB_NATCONN_PREPAREnn and RDB_NATCONN_UPGRADExx Scripts . 1-28	
5.4.15	Persona Nopriv Error Using SQLplus and Other OCI Clients	1-28
5.4.16	Connections to SQL*Net for Rdb Would Hang	1-30
5.4.17	DESCRIBE of an Object Reported Datatype UNKNOWN	1-30
5.4.18	NLS_CHARACTERSET Ignored in ALTER SESSION Command	1-30
5.4.19	Memory Leaks	1-30
5.4.20	Fatal Error When Running Queries Longer Than 16K	1-30
5.4.21	Error Accessing Rdb Database Using Dblink	1-30
5.4.22	Running an Rdb Stored Procedure Using a Database Link	1-31
5.4.23	Protocol Mismatch Error	1-31
5.4.24	SQL*PLUS Hangs Using SQL*Net for Rdb	1-32
5.4.25	Thick JDBC Driver Returned Error	1-32
5.4.26	Select of Column Returned Data in Wrong Format	1-32
5.4.27	Two-phase Commits Using Both Rdb and Oracle Databases	1-33
5.4.28	Logicals Now Defined for ORA_NLS, ORA_NLS32, and ORA_NLS33	1-33
5.4.29	Dblink Failures Fixed	1-33
5.4.30	Better Performance of SQL*Net for Rdb Startup	1-34
5.4.31	Change Maximum CHAR and VARCHAR Sizes	1-34
5.4.32	Fixed Metadata Retrieval Using JDBC	1-34
5.4.33	Access Violations Fixed	1-35
5.4.34	Using Bind Variable Twice in a Query	1-35
5.4.35	ANSI_DATE and SYSDATE Functions Work Correctly	1-35
5.4.36	Wrong Error Returned When Inserting a Duplicate Value into a Unique Index	1-36
5.4.37	SYSDATE Function is Available from a Dblink Connection	1-36
5.4.38	Embedded SQL Program Using SET TRANSACTION Works Correctly	1-36
5.4.39	ALL_TAB_COLUMNS and USER_TAB_COLUMNS Views Return Correct Scale and Precision 1-37	
5.4.40	Insufficient Memory or Quota Problem	1-37
5.4.41	Problem Storing and Retrieving Long Raw Data	1-37

5.4.42	Problem Retrieving Non-English Long Raw Data	1-37
5.4.43	Problem Retrieving Long Raw Data Shorter Than 255 Bytes.....	1-37
5.4.44	Problem Starting Read/Write Transactions.....	1-38
5.4.45	Change to SQL*Net for Rdb Logging	1-38
5.4.46	Updated RDB_NATCONN_XXX_PREPARE.SQL Script.....	1-38
5.4.47	Displayed Release Type Corrected	1-39
5.4.48	Detection of Repeated Intrusion Attempts	1-39
5.4.49	Changes to Scripts.....	1-39
5.4.50	Using Two-Phase Commit from a dblink	1-39
5.4.51	Determining the State of a Transaction When a Two-Phase Commit Fails	1-40
5.4.52	Prestarted Transactions Left Open After a Commit Is Issued	1-40
5.4.53	Using a dblink to a Service Where NLS_LANG Is Defined As Other Than the Default	1-41
5.4.54	Misleading Logging Removed.....	1-42
5.4.55	Schemas Define for Users in the USERS\$ Table	1-42
5.4.56	Using a Bind Variable Twice in a SQL Statement	1-42
5.4.57	Testing for a Read-only Database.....	1-42
5.4.58	FILLM Quota Problem	1-42
5.4.59	NLS_DATABASE_PARAMETERS Added to Scripts	1-43
5.4.60	RDB_NATCONN.COM Problem Fixed	1-43
5.4.61	Problem with Universal Service Fixed	1-43
5.4.62	Memory Leak Related to dblink Fixed	1-44
5.4.63	Connection Problem Fixed.....	1-44
5.4.64	Read Only Transactions Started During Connection.....	1-44
5.4.65	SQL*Plus Invocation Failed with some NSL_LANG Definitions	1-44

List of Tables

1-1	Oracle SQL/Services on OpenVMS Client/Network Support	1-3
4-1	SQL*Net for Rdb Restrictions and Limitations	4-3

List of Examples

4-1	Specifying the OCI Protocol with SQLSRV_MANAGE72	4-4
4-2	Errors Logged to the Dispatcher Log File.....	4-5
4-3	Errors Returned When Accessing DBMS_APPLICATION_INFO	4-6
4-4	Stored Function Module Workaround.....	4-7

Send Us Your Comments

Oracle SQL/Services Release 7.2 Release Notes

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 Oracle Net.

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 OCI Services for Oracle Rdb (formerly known as SQL*Net for Rdb) and should be read to supplement information contained in the *Oracle SQL/Services Installation Guide*, the *Oracle SQL/Services Server 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 OCI Services for Oracle Rdb is included in these release notes (see [Section 1.1.1](#)).

Structure

This manual contains four chapters

- [Chapter 1](#) Describes the new and changed features of Oracle SQL/Services.
- [Chapter 2](#) Describes known software errors fixed in release 7.2 of Oracle SQL/Services and prior releases.
- [Chapter 3](#) Describes problems, restrictions, and workarounds known to exist in Oracle SQL/Services release 7.2.
- [Chapter 4](#) Describes release notes that pertain to OCI Services for Oracle Rdb release 7.2 and prior releases.

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

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

The Oracle Rdb documentation is available from the following web site:

<http://otn.oracle.com/documentation/rdb.html>

Conventions

In this manual, Oracle Rdb refers to Oracle Rdb for OpenVMS software.

HP OpenVMS Industry Standard 64 for Integrity Servers is often referred to as OpenVMS I64.

OpenVMS means both the OpenVMS Alpha and the OpenVMS I64 operating system.

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.

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 beginning with release 7.1. 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 7.2

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

[Section 1.1.1](#) briefly highlights the major changes to the Oracle SQL/Services installation procedure, which are described in detail in the *Oracle SQL/Services Installation Guide*.

The Oracle SQL/Services release 7.2 installation kit contains the following files:

- SQLSRVV72000AM072.A – Oracle SQL/Services for OpenVMS Alpha
- SQLSRVV72000AM072.B – OCI Services for Oracle Rdb for OpenVMS Alpha
- SQLSRVV72000IM072.A – Oracle SQL/Services for OpenVMS I64
- SQLSRVV72000IM072.B – OCI Services for Oracle Rdb for OpenVMS I64
- SQSCLIA072.A – Oracle SQL/Services client kit for OpenVMS Alpha
- SQSCLII072.A – Oracle SQL/Services client kit for OpenVMS I64

The minimum supported version of Oracle Rdb and Oracle SQL/Services client software is V7.0.

1.1.1 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 7.2.

Hardware Requirements

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

Alpha EV56 Requirement

Oracle SQL/Services has been optimized for the Alpha EV56 platform. If running on an older platform, Oracle SQL/Services will still execute properly, but will run some operations in emulation mode and performance will be significantly slower.

DECnet Support for OCI Services for Oracle Rdb Connections

Oracle SQL/Services no longer supports the DECnet network protocol for OCI connections on Alpha platforms.

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 and OCI Services for Oracle Rdb OpenVMS Alpha server.

- OpenVMS Alpha operating system V8.2 or higher
- Oracle Rdb for OpenVMS Alpha V7.0.7 or higher
- One of the following network transport options:
 - Oracle Net

- TCP/IP transport - DEC TCP/IP Services for OpenVMS, 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.

The following software is required for the Oracle SQL/Services and OCI Services for Oracle Rdb OpenVMS I64 server.

- OpenVMS I64 operating system, V8.2-1 or higher
- Oracle Rdb for OpenVMS I64, V7.2 or higher
- One of the following network transport options:
 - DECnet transport - DECnet for OpenVMS I64, V8.2 or higher
 - TCP/IP transport - DEC TCP/IP Services for OpenVMS I64, V5.5-11 or other DEC TCP/IP Services for OpenVMS compliant transport on the host system.
 - Oracle Net

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

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

Desktop	Client Transport Support		
	DECnet	TCP/IP	Oracle 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/Windows ME/Windows XP	X	X	X
Windows 95/Windows 98/ Windows 2000	X	X	X

Table 1–1 (Cont.) Oracle SQL/Services on OpenVMS Client/Network Support

Desktop	Client Transport Support		
Client Platform	DECnet	TCP/IP	Oracle Net
Tru64 UNIX	X	X	X
OpenVMS Alpha	X	X	X
Red Hat Linux	X	X	X

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 XP Client (TCP/IP Transport):

- Microsoft Windows XP Home Edition operating system, V5.1.2600
- Microsoft Windows XP Professional Edition operating system, V5.1.2600

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

- Microsoft Windows ME operating system, V4.90.3000

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

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)

Oracle SQLServices Red Hat Linux client:

- Red Hat Enterprise Linux V2.1 or V3.0

1.2 Running the Oracle SQL/Services Configuration File Utility

If you have release 5.1, 6.0, or 6.1 Oracle SQL/Services configuration files that you want to convert to release 7.2 files, you must first convert the files to release 7.0 files, as shown in [Section 1.2.2](#), and then convert release 7.0 files to release 7.2 files as shown in [Section 1.2.1](#).

1.2.1 Converting Previous Configuration Files to Release 7.2

Follow these steps to convert release 7.0 Oracle SQL/Services configuration files to release 7.2 files:

1. Run the conversion utility:

```
$ @ SYS$MANAGER:SQLSRV_EXT_CONFIG70TO71 sqlsrv_config_file70.dat my_
config.sqs
```

where `sqlsrv_config_file70.dat` is the name of the SQL/Services release 7.0 configuration file that you want to convert and `my_config.sqs` is the SQL/Services manager script file created by this procedure that can be used to create the new configuration file.

If you interrupt the the execution of the command file, check to see if the logical `SQLSRV_EXT_DISP_V70` has been assigned:

```
$ show logical SQLSRV_EXT_DISP_V70
```

If the logical has been assigned, deassign it with the following command before you run the command file again:

```
$ deassign/system SQLSRV_EXT_DISP_V70
```

2. Edit the SQL/Services manager script file (`my_config.sqs`) and make any corrections, such as modifying the log path for the dispatcher or the SQL release number.
3. Follow one of the following procedures to create a configuration file for release 7.2:
 - a. Create a new configuration file:

```
$ SQLSRV_MANAGE72== $SYS$SYSTEM:SQLSRV_MANAGE72.EXE
$ SQLSRV_MANAGE72
SQLSRV> set config_file temp.dat;
The configuration file does not exist or could not be opened.
Create it now? (N)
```

```
SQLSRV>@MY_CONFIG.SQS;
SQLSRV>EXIT
$ COPY TEMP.DAT;SYS$MANAGER:SQLSRV_CONFIG_FILE72.DAT;
$ @SYS$STARTUP:SQLSRV$STARTUP72
```

- b.** Or follow these steps to add the necessary dispatchers and services to the configuration file created by the installation:

- Edit the my_config.sqs file to remove lines such as creating a server.
- Run the following commands:

```
$SQLSRV_MANAGE72 ::= $SYS$SYSTEM:SQLSRV_MANAGE72.EXE
$SQLSRV_MANAGE72
SQLSRV> @MY_CONFIG.SQS; -- This file must be modified before it can
be run
SQLSRV> EXIT
$ @SYS$STARTUP:SQLSRV$STARTUP72
```

1.2.2 Converting Previous Configuration Files to Release 7.0

In order to convert release 5.1, 6.0, or 6.1 Oracle SQL/Services configuration files to release 7.0 files, you can run the conversion utility, SYS\$MANAGER:SQLSRV_CONVERT_CONFIG.EXE. This conversion utility converts any Oracle SQL/Services release 5.1, 6.0, or 6.1 configuration file into a SQLSRV_MANAGE script file that you can then run to add your previously defined classes to a release 7.0 configuration file as services. You can run the utility by following these steps:

- Run the image, taking defaults for the input and output file names. The defaults are:
SYS\$STARTUP:SQLSRV\$CONFIG.DAT
SYS\$STARTUP:SQLSRV_CONFIG.SQS
- Define logical names to accept an input file name and produce an output file name from specific directories. For example:

```
$ DEFINE SQLSRV$CONFIG SYS$MANAGER:SQLSRV$CONFIG61.DAT
! for the existing SQLSRV$CONFIG61.DAT
$ DEFINE SQLSRV_CONFIG MY_CONFIG_SCRIPT_FILE.SQS
! for new MY_CONFIG_SCRIPT_FILE.SQS
$ RUN SYS$MANAGER:SQLSRV_CONVERT_CONFIG.EXE
```

- Define a foreign symbol using the arguments p1 and p2 as the input and output file names. For example:

```
$ CVT ::= $SYS$MANAGER:SQLSRV_CONVERT_CONFIG.EXE
```

```
$ CVT SYS$MANAGER:SQLSRV$CONFIG61.DAT MY_CONFIG_SCRIPT_FILE.SQS
```

The correct Oracle Rdb version must be set for each service (CLASS in terminology used by releases prior to release 7.0). In previous versions of Oracle SQL/Services, the version was set by the login procedure of the account specified as the STARTUP account in the configuration file. In Oracle SQL/Services release 7.0 and higher, the version is an attribute (SQL VERSION) of the defined service. The conversion utility attempts to default to this version depending on the name of the STARTUP account; however, the default may not be correct. Before creating the script file entry, the conversion utility displays the selected default SQL VERSION for the service and allows you to take the default or enter a different version. The version may be either the string STANDARD or x.x where x is a decimal digit.

Once you have created the SQLSRV_MANAGE script file, you can run it to add the services to your release 7.0 configuration file. For example:

```
$ SQLSRV_MAN ::= $SYS$SYSTEM:SQLSRV_MANAGE.EXE  
$ SQLSRV_MAN -INPUT MY_CONFIG_SCRIPT_FILE.SQS
```

1.3 Documentation

Documentation for Oracle SQL/Services and OCI Services for Oracle Rdb is available in Adobe Acrobat (PDF) formats. Adobe Acrobat files ending with extension .PDF can be read with an Adobe Reader. Readers for many platforms are available without fee from the Adobe web site.

Follow this link to access documentation on MetaLink:

```
Top Tech Docs\Database\Rdb:Sql/Services\Documentation Tab\  
SQL/Services...Documentation Index
```

Follow this link to access documentation on OTN:

```
www.oracle.com\Rdb\Documentation\Oracle SQL\Services Documentation
```

1.4 Summary of Oracle SQL/Services New and Changed Features for Release 7.2

The following sections describe new and changed features for Oracle SQL/Services release 7.2. See [Chapter 4, "OCI Services for Oracle Rdb: Release Notes"](#), for descriptions of new and changed features for OCI Services for Oracle Rdb release 7.2.

1.4.1 Oracle SQL/Services Available on OpenVMS I64

Oracle SQL/Services release 7.2 is available on the HP OpenVMS Industry Standard 64 for Integrity Servers operating system.

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:

- Macintosh for MPW API

The Macintosh for MPW client API is deprecated in Oracle SQL/Services release 7.1.5 and higher, 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 or higher kits.

- Macintosh for THINK C Client API

The Macintosh for THINK C client API is deprecated in Oracle SQL/Services release 7.1.5 and higher, 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 or higher kits.

- Microsoft Windows 3.1 client API

The Microsoft Windows 3.1 client API is deprecated in Oracle SQL/Services release 7.1.5 and higher, 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 or higher kits.

- Solaris client API

The Solaris client API is deprecated in Oracle SQL/Services release 7.1.5 and higher, 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 or higher kits.

- MS-DOS large memory model client API

The MS-DOS large memory model client API is deprecated in Oracle SQL/Services V7.0 and higher, 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 or higher kits.

- 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 higher 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 higher 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 higher 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 higher 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 higher 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.

Oracle SQL/Services: Software Errors Fixed

This chapter describes problems with Oracle SQL/Services software that are fixed in this release.

2.1 Oracle SQL/Services Errors Fixed in Release 7.2

The following known problems found in the Oracle SQL/Services OpenVMS server have been fixed for release 7.2.

2.1.1 Monitor Aborts When Connection Cancelled

If you attempted to use the monitor port for an OCI connection and then cancelled the process, the monitor would abort.

This problem has been fixed for this release.

Oracle SQL/Services: Known Problems

This chapter describes problems and restrictions relating to Oracle SQL/Services release 7.2.

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 7.2 Server Known Problems and Restrictions

The following sections describe Oracle SQL/Services release 7.2 server restrictions and known problems.

3.2.1 Support for OpenVMS VAX and Standard Kits

Oracle SQL/Services release 7.2 and OCI Services for Oracle Rdb release 7.2 are not supported on OpenVMS VAX.

Also, standard kit installation is not supported with this version. Only multiversion kits are available.

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 SYS\$SETDDIR 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 fails 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 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.

3.2.4 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.

3.2.5 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$default';
syntax error at ''Owner', required parameter not specified'
```

3.2.6 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.

3.2.7 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.

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

Monitor, dispatcher log, and bugcheck dump files are written to the SYSS\$MANAGER 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.

3.2.9 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 SYSS\$SYSTEM:loginout image with a process-specific command procedure as SYSS\$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.

3.2.10 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.

3.2.11 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:
  DEcnet 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 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.

3.2.12 How to Re-create 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:
    DECnet 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 SYSS\$MANAGER:SQLSRV_CREATE[70].COM command procedure. This procedure invokes the SYSS\$MANAGER: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 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;
```

3.3 Oracle SQL/Services Release 7.2 Client Known Problems and Restrictions

The following information describes Oracle SQL/Services release 7.2 client known problems and restrictions.

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

The Oracle SQL/Services client shared image for OpenVMS is now compiled using HP C. The options file provided by Oracle SQL/Services for linking client applications has changed. It used to include SYSS\$LIBRARY:VAXCRTL\$API/SHARE. It now includes SYSS\$LIBRARY: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 SYSS\$LIBRARY:VAXCRTL/SHARE and link against this new options file as well as SYSS\$LIBRARY: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 section 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.

- 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*

The Guide 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.

OCI Services for Oracle Rdb: Release Notes

This chapter highlights release notes that pertain to OCI Services for Oracle Rdb (formerly known as SQL*Net for Rdb). It contains information about installation, new and changed features, known problems, software fixes, and documentation changes.

4.1 Software Requirements

OCI Services for Oracle Rdb release 7.2 requires OpenVMS Alpha Version 8.2 or higher, or OpenVMS I64 Version 8.2-1 or higher software.

4.2 Installing OCI Services for Oracle Rdb

The installation for OCI Services for Oracle Rdb is part of the installation for Oracle SQL/Services release 7.2. Refer to the following documentation for information on installing OCI Services for Oracle Rdb.

- `SYSS$HELP:SQLSRV072_INSTALL_GUIDE.PDF`

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

4.2.1 Problem Reporting

If an error occurs while you are using OCI Services for Oracle Rdb and you believe that the error is caused by a problem with this Oracle product, contact your Oracle 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 OCI Services for Oracle Rdb session. By providing

the logged information with your problem report, you supply important data that can help solve the problem. See *Oracle SQL/Services Server Configuration Guide* for more information about using the SQL ALTER SESSION LOG statement.

4.3 Upgrading Oracle Rdb Release 7.0 Database to Oracle Rdb Release 7.1

Oracle recommends that you upgrade your database from Oracle Rdb release 7.0 to Oracle Rdb release 7.2, and then upgrade the release 7.2 database using the RDB_NATCONN72 command file.

4.4 New and Changed Features for OCI Services for Oracle Rdb Release 7.2

The following sections describe new or changed features for OCI Services for Oracle Rdb release 7.2.

4.4.1 OCI Services for Oracle Rdb Available on OpenVMS I64

OCI Services for Oracle Rdb release 7.2 is available for the HP OpenVMS Industry Standard 64 for Integrity Servers operating system.

4.5 Known Problems and Restrictions

This section highlights problems and restrictions relating to OCI Services for Oracle Rdb and includes workarounds where appropriate.

4.5.1 Support for OpenVMS VAX and Standard Kits

Oracle SQL/Services release 7.2 and OCI Services for Oracle Rdb release 7.2 are not supported on OpenVMS VAX.

Standard kit installation is not supported with this version. Only multiversion kits are available.

4.5.2 Restrictions and Limitations

The restrictions and limitations in [Table 4–1](#) apply to this release of OCI Services for Oracle Rdb:

SQL Functions Return Incorrect User Information

Table 4-1 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>Oracle SQL/Services Server Release 7.1.6 Configuration Guide</i> for information about using other character sets.
Dates	Oracle dates can pre-date the OpenVMS 17-NOV-1858 date. Such dates are not supported by Oracle Rdb or OCI Services for Oracle 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	OCI Services for Oracle Rdb does not support multischema Oracle Rdb databases. However, OCI Services for Oracle 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, OCI Services for Oracle 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	OCI Services for Oracle Rdb sets the Oracle_Level1 or Oracle_Level2 dialect and requires the dialect to perform its tasks. Changing the dialect may result in unexpected OCI Services for Oracle Rdb failures.
Multisession server is not supported	The Oracle server can support multiple sessions from multiple client applications in one server process. OCI Services for Oracle 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.

Table 4–1 (Cont.) SQL*Net for Rdb Restrictions and Limitations

Category	Description
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. OCI Services for Oracle 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 <code>SELECT col1 FROM tab1</code> were parsed and described in OCI, the Oracle server returns a message that col1 either does or does not have a NOT NULL constraint. OCI Services for Oracle 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.

4.5.3 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_MANAGE72

```
$ MCR SQLSRV_MANAGE72
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-SQSMGERR, 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.

4.5.4 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 7.2.

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 7.2 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 7.2 configuration. If an existing release 7.2 configuration is found, the installation procedure will not change the RMU dispatcher, so you should alter the RMU dispatcher to disable autostart.

4.5.5 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 OCI Services for Oracle 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.

4.5.6 Oracle Reports: Problem with release 7.2 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 release 7.2 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.

4.5.7 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.

4.5.8 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 an OpenVMS system try to link debug images in the nondeferrable mode, it will fail. This mode does not currently work on OpenVMS systems.

4.5.9 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".

Previous Releases: New Features and Fixed Problems

This chapter describes the new features and technical changes to previous releases of Oracle SQL/Services and OCI Services for Oracle Rdb. It also describes problems that were fixed in these products.

5.1 New and Changed General Features in Previous Releases of Oracle SQL/Services

5.1.1 New Logical `SQLSRV$MAX_EXECUTOR_FAILURES`

In Oracle SQL/Services releases prior to release 7.1.6, an executor would fail if two startup failures occurred. If there were no other active executors for the service at the time of the second failure, the service would also be shut down.

For example, if a database was closed for maintenance, any user who attempted to connect to that database through a service would get an executor startup failure. After two such failed users through an executor of that service, the executor would be shut down.

In Oracle SQL/Services release 7.1.6, support for a new system logical was added to allow users to control the maximum number of executor failures allowed before an executor fails. The logical `SQLSRV$MAX_EXECUTOR_FAILURES` can be defined to change the maximum failures from the default of 2 to any positive integer value. In this way, a user can control how often executors and services shut down during routine database maintenance.

By defining the following with Oracle SQL/Services release 7.1.6 or later, a user will ensure that an executor will not be terminated until at least 10 failures have occurred.

```
$ define SQLSRV$MAX_EXECUTOR_FAILURES 10
```

5.1.2 Logical SQLSRV\$LOG_CONNECTIONS

Beginning with release 7.1.6, you can define the logical SQLSRV\$LOG_CONNECTIONS to "NO" so that successful connections are not logged to dispatcher log files, and the size of the dispatcher log files is reduced. If the logical is undefined or assigned to any other value, the successful connections are logged. Because this logical is evaluated when a dispatcher is started, the dispatcher must be restarted if the logical is changed.

5.1.3 Oracle Server Two-Phase Commit Support

Oracle SQL/Services release 7.1.6 provides two-phase commit support with the following capabilities:

- Oracle Rdb databases can fully participate in Oracle RDBMS-managed distributed transactions.
- Multiple Oracle server DBLINKs to Rdb databases can participate in a transaction.
- The DECdtm XA Gateway provides an interface between the Oracle distributed transaction protocol and DECdtm distributed transaction protocol.

Perform the following steps to enable the two-phase commit protocol:

- Define the following logical in your service process initialization file:

```
DEFINE RDB$DDTM_XG_INFO gateway-name
```

where gateway-name is the name specified in the CREATE_LOG/GATEWAY_NAME command in XGCP, the XA Gateway control program.

- Add the following command to the SQL Initialization file used by the CREATE SERVICE and ALTER SERVICE commands:

```
ALTER SESSION SET TX_MODE NOWARN_1PC
```

- Define the following logical:

```
DEFINE SQLNET_RECO_USER "username/password"
```

Note: The two-phase commit functionality currently does not work with DECdtm XA Gateway Version 2.1, which is installed with OpenVMS Version 7.3-2 and later. The functionality will be available with later versions of XA Gateway when an update for XA Gateway or DECdtm is released. If you try to use two-phase commit with XA Gateway Version 2.1, the database and the XA Gateway can become locked when the process tries to do a recover for a failed transaction, and manual intervention is required.

5.1.4 Linux Client Available

Beginning with release 7.1.6, Oracle SQL/Services supports the Linux client as well as Windows, HP Tru64 Unix, and OpenVMS clients. For prerequisite and installation information for the Linux client, read the `Linux_client_readme.txt` file included with the Linux kit, found in `sqllinuxcli.tar`. The Linux client kit is available on the CD for the Oracle Rdb client kit.

5.1.5 Executor Failure Maximum Logical

In Oracle SQL/Services releases prior to 7.1.6, an executor would fail if two startup failures occurred. If there were no other active executors for the service, at the time of the second failure, the service would also be shut down.

For example, if a database was closed for maintenance, any user who attempted to connect to that database through a service would get an executor startup failure. After two such failed users through an executor of that service, the executor would be shut down.

In Oracle SQL/Services release 7.1.6, support for a new system logical has been added to allow users to control the maximum number of executor failures allowed before an executor fails. The logical `SQLSRV$MAX_EXECUTOR_FAILURES` can be defined to change the maximum failures from the default of two to any positive integer value. In this way, you can control how often executors and services shut down during routine database maintenance.

For example, issue the following command to ensure that an executor will not be terminated until at least ten failures have occurred:

```
$ define SQLSRV$MAX_EXECUTOR_FAILURES 10
```

5.1.6 Enhancements to SQLSRV\$DEINSTALL_DELETE Procedure

The SQLSRV\$DEINSTALL_DELETE.COM procedure no longer removes the SQLNET4RDB identifier. If the command procedure is executed on a cluster, it now displays information about all SQL/Services monitors that are currently executing on the cluster prior to asking if the user would like to continue, as shown in the following example:

```
$ @SQLSRV$DEINSTALL_DELETE

Oracle SQL/Services versions currently installed on your system

1 Version 7.1 (Multiversion)
0 Quit

Enter Choice to deinstall (0...1) : 1

You are about to deinstall Oracle SQL/Services 7.1

This procedure will delete SYSSMANAGER:SQLSRV$SHUTDOWN71.COM.
If the Oracle SQL/Services 7.1 monitor is running on any
other node on your cluster besides the node SYS1,
you will have to manually stop the monitor on each of these other
nodes after this procedure has finished.

The following monitor(s) are currently executing on your cluster.

OpenVMS V7.3-2 on node SYS2 4-JUN-2004 14:34:23.61 Uptime 10 01:23:29
Pid Process Name State Pri I/O CPU Page flts Pages
20400126 sqlsrv_mon_0071 HIB 6 2020 0 00:00:00.79 399 523

OpenVMS V7.3-2 on node SYS3 4-JUN-2004 14:34:23.63 Uptime 19 20:39:17
Pid Process Name State Pri I/O CPU Page flts Pages
2080025D sqlsrv_mon_0071 HIB 6 599 0 00:00:25.98 1321 3030

Enter Y(ES) to continue to deinstall Oracle SQL/Services 7.1 :
```

5.1.7 New Logical Name

A new logical name was added in release 7.1.5.6 to allow you to disable the service log for all services:

```
$DEFINE/SYSTEM SQLSRV_EXEC_LOG NOLOG
```

This logical must be defined before a service is started. Oracle Corporation recommends that you do not disable the service log because it is needed if a problem occurs.

5.1.8 Version Check

Beginning with release 7.1.5.6, if you try to use a configuration file created by Oracle SQL/Services prior to release 7.1.5.4, you will get a bugcheck dump and an error message will be written to the monitor log file.

5.1.9 New and Changed SQLSRV_MANAGE Features

The following new capabilities and logical names have been added:

5.1.9.1 Recall function

You can now use the up arrow and down arrow keys at the SQLSRV> prompt to recall any of the last 20 commands that were entered.

5.1.9.2 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]
```

5.1.9.3 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 SYS\$MANAGER directory.

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

5.1.9.4 Extract Command

The EXTRACT command was added in release 7.1.5.6. It extracts definitions that are written to an Oracle SQL/Services command script. The script can be used to re-create servers, dispatchers, and services.

FORMAT

```
EXTRACT keyword [option]
```

Keyword can be one of the following:

- Server
Extracts the definition for the server. There is no option for this keyword.
- Disp [disp_name]
If the disp_name is omitted, definitions for all dispatchers are extracted. If the disp_name is specified, the definition for just that dispatcher is extracted.
- Service [service_name]
If the service_name is omitted, definitions for all services are extracted. If the service_name is specified, the definition for just that service is extracted.

EXAMPLES

The following example extracts the definition for the service OCI_AAA and displays the information on the output device.

```
$ SQLSRV_MANAGE71
SQLSRV> extract service OCI_AAA;
Create Service OCI_AAA
  Owner                'AAA'
  Protocol              OCI
  SQL version          7.1
  Autostart            off
  Process_initialization 'DBD_USER6:[JONES]INIT_OCI_ENG70.COM'
  ATTACH 'filename DBD_USER6:[JONES]mf_personnel'
  Reuse scope is       SESSION
  Database Authorization CONNECT USERNAME
  SQL_init_file        'DBD_USER6:[JONES]init.sql'
  Application Transaction Usage SERIAL
  Idle Executor Timeout 1800
  Min Executors        1
  Max Executors        10
  Clients Per Executor 1
;
Grant use on service OCI_AAA
  To 'AAA'
;
SQLSRV>
```

The following example extracts definitions for the entire dispatcher definition and writes them to an output file.

```
$ SQLSRV_MANAGE71 -OUTPUT A.SQL
SQLSRV> extract disp;
SQLSRV> exit
```

```
$TYPE A.SQL
Create Dispatcher SQLSRV_DISP
  Autostart                off
  Max connects              100
  Idle User Timeout        0
  network_port DECnet object 81          protocol SQLServices
  network_port tcpip port_id 118        protocol SQLServices
  Log path                  'SYS$MANAGER:'
  Dump path                 'SYS$MANAGER:'
  ;
```

5.1.9.5 ALTER DISPATCHER and CREATE DISPATCHER

The ALTER DISPATCHER and CREATE DISPATCHER commands have been enhanced to include LOG PATH and DUMP PATH arguments. These arguments allow you to specify a directory name for log and bugcheck dump files, for example:

```
SQLSRV> ALTER DISPATCHER SQLSRV_DISP1 LOG PATH 'USER1:[SQLSRV_TEST1.AAA]
_SQLSRV> DUMP PATH 'USER1:[SQLSRV_TEST2.BBB]';
%SQLSRV-S-ALTER_RESTART, Restart object to have altered settings take effect
```

If you specify NOLOG instead of a directory name for the LOG PATH qualifier, no log file is written, for example:

```
SQLSRV> CREATE DISPATCHER SQLSRV_DISP1 LOG PATH 'NOLOG'
_SQLSRV> DUMP PATH 'SYS$MANAGER';
```

Oracle recommends that you create a log file for troubleshooting purposes unless you have a problem with excessive I/O entries in the log file. The default directory for both qualifiers is SYS\$MANAGER.

The LOG PATH and DUMP PATH arguments are available when you use the SQLSRV_MANAGE command utility, but not when you use the SQLSRV_MANAGE GUI tool.

If values are assigned to existing logicals `SQLSRV_DISP_LOGPATH` and `SQLSRV_DISP_DUMPPATH`, they override log path and dump path values specified by the `LOG PATH` and `DUMP PATH` arguments.

The `SHOW DISPATCHER` command has been enhanced to display the values for Log path and Dump path:

```
SQLSRV> SHOW DISP SQLSRV_DISP;
Dispatcher SQLSRV_DISP
.
.
.
Log Path:          USER1:[SQLSRV_TEST1]
Dump path:         USER1:[SQLSRV_TEST1]
.
.
.
** This Dispatcher will be updated as follows when it is restarted **
Log path:          USER1:[SQLSRV_TEST1.AAA]
Dump path:         USER1:[SQLSRV_TEST2.BBB]
```

5.1.9.6 ALTER DISPATCHER and CREATE DISPATCHER

- * New argument: `PROTOCOL [NATIVE | OCI | SQLSERVICES]`

5.1.9.7 ALTER SERVICE and CREATE SERVICE

- * New argument: `PROTOCOL [OCI | RMU | SQLSERVICES]`
- * New argument: `DEFAULT_CONNECT_PASSWORD <password-string>`
- * New argument: `OWNER_PASSWORD <password-string>`

5.1.9.8 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.

5.1.9.9 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.

5.1.9.10 SHOW

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

5.1.10 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.
- **Windows 98 and Windows 2000**
Windows 98 and Windows 2000 are now supported operating systems for Oracle SQL/Services release 7.1.6.

5.1.11 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.

5.2 Oracle SQL/Services Errors Fixed in Prior Releases

The following known problems found in the Oracle SQL/Services OpenVMS server in previous releases have been fixed.

5.2.1 Shared Memory Not Released with Continuous Start and Shutdown of Server

Free shared memory would decrease to zero if services were started and stopped continuously. This has been fixed in release 7.1.6.

5.2.2 Erroneous RDB\$_NO_PRIV Errors Using OCI Services

In releases prior to 7.1.5.9.1, users would sometimes get erroneous RDB\$_NO_PRIV messages connecting to an OCI service. This problem was fixed in Oracle SQL/Services and SQL*Net for Rdb release 7.1.5.9 and Oracle Rdb release 7.0.7.2.

5.2.3 OCI Universal Services Not Correctly Impersonating the Connect User

In Oracle SQL/Services V7.1.5.8, OCI universal services with database authorization defined as CONNECT USER were not correctly impersonating the connect user. Therefore, Rdb external functions would be executed under the service owner, rather than the connect user account.

5.2.4 Long Queries Hang

Oracle SQL/Services release 7.1.5.8 introduced a problem that could cause long queries from an OCI source to sometimes hang. Examination of the SQL/Services processes would show that they were idle even though the client had not received a response from SQL/Services.

This problem would occur when a network packet from the client contained more than one record. The SQL/Services dispatcher would process the first record and then wait for notification of the next record. Since the next record was already in its buffer no further notification would be delivered. The SQL/Services dispatcher would never process the next record in its buffer and the client application would hang.

There was no workaround for this issue.

The SQL/Services dispatcher now explicitly checks to see if there are any additional messages in the network buffer before entering its idle loop.

5.2.5 Poor Performance From OCI Queries

SQL/Services Release 7.1.5.8 introduced a problem that could cause queries from an OCI source to sometimes take an extraordinarily long time to complete. Examination of the SQL/Services processes would show that they were idle even though the client had not received a response from SQL/Services.

This problem was introduced by changes in the underlying SQLnet libraries employed by SQL/Services. There were instances where a network message would arrive but notification of that arrival was not being delivered to the SQL/Services dispatcher. When that occurred SQL/Services would only see that a new network message had arrived when it did periodic polling.

There was no workaround for this issue.

Notification of network message arrival is now done immediately.

5.2.6 Monitor Crashes with "bind mon: unexpected monitor state"

Under rare timing conditions it was possible for the SQL/Services monitor to crash with a bugcheck stack trace similar to the following:

```
%DBS-F-BUGCHECK: in DBS01:[SQLSRV_BUILD.NOV_05_2003.DBS_SRC]CMD.C:1 at line
11003 bind mon: unexpected monitor state
```

This problem would occur when multiple SQLSRV_MANAGE clients were sending "system management" requests to the SQL/Services monitor. If one of those clients sent a request that required many buffers of data for the monitor to respond, and that client was slow in reading the messages sent by the monitor, causing all available buffers for that client to be consumed, the monitor would process messages from other clients until the slow client had read its buffers. If the monitor had stalled on a slow client process, and a new client connection "CONNECT SERVER command" arrived while the monitor was stalled on the slow process, a bugcheck dump would occur and the monitor process would fail.

The bugcheck would occur because the routine that handled new connection requests assumed that there could not be a current client connection active when processing a new connection request. When this routine saw that condition it would force a bugcheck and monitor termination. However, this condition is possible and acceptable and thus there is no reason to bugcheck.

This bugcheck exception has been removed.

The incidence of this problem can be reduced if only one SQLSRV_MANAGE process is run at one time.

5.2.7 Increased ENQLM Quota

Beginning with OpenVMS Version 7.1, any process that has its ENQLM set to 32767 is granted almost unlimited ENQLM. With this release, SQL/Services processes have this capability. If the service owner has its ENQLM set to 32767, the service process can have ENQLM of 16776959.

5.2.8 Severity of Error Changed

The severity of the error SQLSRV-E-FLOBEXECUNAVAIL has been changed from -E- to -I- to reflect its correct status as an informational message.

5.2.9 SERVICE_NAME Now Allowed in Tnsnames.ora

Beginning with release 7.1.5.2, the logical SERVICE_NAME has been allowed in TNSNAMES.ORA, for example:

```
cra_univ1 =
  (DESCRIPTION =
    .
    .
    .
    (CONNECT_DATA =
      (SERVICE_NAME = oci_eng_univ_cra
        .
        .
        .
      )
    )
  )
```

5.2.10 Installation Procedure Overwrote Configuration File

In previous versions of SQL/Services the installation procedure overwrote the configuration file during installation.

This problem has been fixed if you are upgrading from multiversion to multiversion, which is the only version that has been supported since SQL/Services release 7.1.5.

If you are upgrading from a release 7.1.* standard version to release 7.1.5.4 or later, the installation still overwrites the configuration file. To work around this problem: follow these steps prior to running the installation procedure:

- Rename the configuraton file or move it to another location.
- Run the SYSS\$MANAGER:SQLSRV\$DEINSTALL_DELETE.COM command file (the routine that overwrites the configuration file).
- If you are upgrading from release 7.1 or later, move the configuration file back to its original location or rename back to its original name and append '71' to the end of the filename.
- Run VMSINSTAL to finish the upgrade to 7.1.5.6.
- If you are upgrading from a release prior to 7.1, conversion of configuration files is not supported. The installation procedure creates a basic configuration file which you can modify by using the SQLSRV_MANAGE utility.

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

5.2.11 Preattached Database Service with Default_connect_username Could Bugcheck

In previous versions, if you created a preattached database service and included a default_connect_username, the dispatch could bugcheck with an access violation when you attempted to connect. This problem could occur on VAX as well as Alpha machines.

5.2.12 Impersonation Is Supported

PERSONA support was a new feature in Oracle Rdb release 7.1.0.0. This feature enables the full impersonation of an OpenVMS user. Impersonation is now supported in Oracle SQL/Services.

5.2.13 Logicals Now Defined for ORA_NLS, ORA_NLS32, and ORA_NLS33

In earlier versions of SQL/Services, users that wanted to define a language other than AMERICAN had to define the logical names ORA_NLS, ORA_NLS32, and ORA_NLS33 to point to message files and other NLS files.

SQLSRV_MON71.EXE now contains definitions for all the ORA_NLS nn logical names and users no longer need to define them.

5.2.14 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.

5.2.15 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.

5.2.16 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.

5.2.17 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 SLOCTET_LEN value.

5.2.18 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.

5.2.19 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.

5.2.20 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.

5.2.21 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, `SQLSRV$$SHUTDOWN.COM` and `SQLSRV$$STARTUP.COM` have been improved to check for and terminate dangling Oracle SQL/Services processes.

5.2.22 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.

5.2.23 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.

5.2.24 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.

5.2.25 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 was fixed in V7.0.2.

5.2.26 Upgrade of Oracle SQLNET Libraries

Releases of Oracle SQL/Services prior to 7.1.5.8 included Oracle SQLNET release 2.1.5 libraries. Oracle SQL/Services release 7.1.5.8 has been upgraded to include Oracle SQLNET release 9.2.0.2 libraries. Any restrictions or problems experienced using Oracle SQL/Services that were due to Oracle SQLNET release 2.1.5 library problems, which have subsequently been fixed in Oracle SQLNET release 9.2.0.2, have now been solved.

As a result of this upgrade, the restriction of a maximum of 30 concurrent connections for an OCI service has now been lifted for Oracle SQL/Services release 7.1.5.8.

5.2.27 Memory Leak of Monitor Process

The memory leaks of the monitor process with the following commands have been fixed:

```
SQLSRV> CONNECT SERVER;  
SQLSRV> DISCONNECT SERVER;  
SQLSRV> SHOW commands;  
SQLSRV> EXTRACT commands;  
SQLSRV> DROP commands;  
SQLSRV> ALTER commands;
```

5.2.28 Executing External Routines from Universal OCI Services

With Oracle SQL/Services releases prior to release 7.1.5.8, universal OCI services would sometimes require that the connect user have SYSPRV or GRPPRV in order to successfully execute an external routine.

This problem has been fixed in Oracle SQL/Services release 7.1.5.8.

5.2.29 Support for Multiline Statements in SQL Initialization File

Prior to SQL/Services release 7.1.5.8, SQL statements in the SQL initialization file were required to be on one line, and were restricted to a length of 512 characters. This restriction has been removed - multiline statements are now supported in the SQL initialization file.

A hyphen must be used as a continuation character at the end a line in the initialization file to indicate that the SQL statement continues on the following line. The limit of length for one line in the initialization file is 512 characters, so if the SQL statement exceeds 512 characters, you must use additional lines.

5.3 New and Changed Features for Previous OCI Services for Oracle Rdb Releases

This section highlights new and changed features for previous OCI Services for Oracle Rdb releases.

5.3.1 Date-Time Data Type Enhancements

The following date-time data types have been implemented:

- **TIME** data type
If structured date datatypes have been turned on, columns can be defined as type **TIME**.
- **INTERVAL DAY-TO-SECOND** data type
If structured date datatypes have been turned on, columns can be defined as type **INTERVAL DAY <size> TO SECOND <size>**.
- **INTERVAL YEAR-TO-MONTH** data type
If structured date datatypes have been turned on, columns can be defined as type:
 - **INTERVAL YEAR<size> TO MONTH <size>**
 - **INTERVAL YEAR<size>**
 - **INTERVAL MONTH <size>**

Two logical names have been added in release 7.1.6:

- **SQLNET_TIMESTAMP_DATE_TYPE**
You can define this logical to 'Y' or 'y' in order to use the **TIMESTAMP** datatype.
- **SQLNET_STRUCTURED_DATE_TYPES**
You can define this logical to 'Y' or 'y' in order to use the structured date data types.

5.3.2 ALTER SESSION Enhancements

The **ALTER SESSION** statement accepts the following new syntax:

- **ALTER SESSION SET SQLNET_TIMESTAMP_DATE {ON | OFF}**
Allows the interactive ability to turn on and off the usage of **TIMESTAMP** data types. The result is the same as setting the **SQLNET_TIMESTAMP_DATE_TYPE** logical.
- **ALTER SESSION SET SQLNET_STRUCTURED_DATE_TYPES {ON | OFF}**
Allows the interactive ability to turn on and off the usage of structured date types. The result is the same as setting the **SQLNET_STRUCTURED_DATE_TYPES** logical.
- **ALTER SESSION SET SQLNET_DEBUG_FLAGS {values}**
Allows the interactive ability to set the debug flag values for debug logging purposes. The result is the same as setting the **SQLNET_DEBUG_FLAGS** logical.

5.3.3 Emulate Oracle release 9.2.0.4

OCI Services for Oracle Rdb release 7.1.6 emulates Oracle release 9.2.0.4.

5.3.4 Dictionary Prepare/Upgrade/Drop Program Replaces Scripts

In release 7.1.6 and later of OCI Services for Oracle Rdb, the RDB_NATCONN_PREPARE/UPGRADE/DROP scripts have been replaced by a program, RDB\$NATCONN_DICnn. This program prepares a database that has not been previously prepared, upgrades a database previously prepared at release 1.0.2.5 and later to the current version, and drops all NATCONN objects created by any version of the scripts. The RDB\$NATCONN_DICnn program is invoked by the RDB\$NATCONNnn.COM command file. For information on using the command file, see the *Oracle SQL/Services Installation Guide*.

5.3.5 More Efficient Dictionary Queries

This release of OCI Services for Oracle Rdb has changed the way many of the dictionary queries are done, both to Oracle Rdb tables and to the NATCONN tables. This may be especially apparent in faster connect/attach times.

5.3.6 Hidden Objects in OCI Services for Oracle Rdb

The OCI Services for Oracle Rdb data dictionary program, RDB\$NATCONN_DICnn.EXE, makes use of a feature, first available in Oracle Rdb release 7.1.3, that defines all the OCI Services for Oracle Rdb objects created by the program as hidden objects. They will not appear in the output of a SHOW TABLES command; if you want to see them you must enter the command SHOW SYSTEM TABLES (or SHOW SYSTEM MODULES, etc).

If you are running a version before release 7.1.3, the behavior will be the same as it is currently; that is, tables, modules, etc., will appear as user objects in the database, rather than as system objects. If you run the upgrade option rather than the prepare option, you will get a mixture of hidden and not hidden objects, depending on what version you are upgrading from. In that case, you must enter a SHOW TABLES and a SHOW SYSTEM TABLES command to see all the OCI Services for Oracle Rdb tables. Oracle recommends that you drop all OCI Services for Oracle Rdb objects from your database, and use the prepare option. You can save usernames and passwords from your USER\$ table in a new table, then prepare the database, and insert the usernames and passwords back into the new USER\$ table.

5.3.7 New Tables in Oracle Data Dictionary

The following three tables and views are created by the OCI Services for Oracle Rdb data dictionary program: DBA_USERS, DBA_DB_LINKS, and DBA_SYS_PRIVS. These tables and views did not get created in earlier versions of OCI Services for Oracle Rdb.

5.3.8 Enhanced RDB_NATCONNnn.COM File

Beginning with release 7.1.6, the RDB_NATCONNnn file can be used to modify, drop, and show users as well as add them. See the *Oracle SQL/Services Server Configuration Guide* for information about how to use these capabilities.

5.3.9 Optional validation of connecting program name

New functionality in this release allows sites to restrict which programs are allowed to access each database through OCI Services for Oracle Rdb. When the database is prepared or upgraded for OCI Services, a new table, ORA_VALID_PROGRAMS, is created. It has two columns, USERNAME and PROGRAM. These columns are used in a LIKE comparison to validate the user and program that are connecting. To activate this functionality you must define a new logical SQLNET_VALIDATE_PROGRAM as Y or y in the process initialization file for the service. The table ORA_VALID_PROGRAMS allows select access to public but insert, update, and delete only to SQLNET4RDB. If the logical is defined, OCI Services checks at connection time that there is an entry in the ORA_VALID_PROGRAMS table that matches the user and program that are connecting. Entries in the table must use the syntax of a LIKE comparison; that means that an entry of '%' in the USERNAME column would allow any user. An entry of %SQLPLUS% in the PROGRAM column would allow SQLPlus from any platform. Both columns of ORA_VALID_PROGRAMS must contain data for each row. An entry of '%' in both columns would allow any user from any program to connect; not defining the logical SQLNET_VALIDATE_PROGRAM has the same effect. All other validation and security checking is still done; this will NOT allow anyone access to the database without all required privileges. It can only restrict usage by some or all users to a particular program or set of programs.

The logical SQLNET_VALIDATE_PROGRAM can be defined as Y or y to turn on program validation, so that the program name is validated against the program name and username entries in the ORA_VALID_PROGRAMS table. This definition allows clients that do not send their program name to connect. In order to disallow null program names, define the logical as "NONULL" or "nonnull". The program name check is case sensitive, so it may be necessary to include an entry for %SQLPLUS% and %sqlplus% in the ORA_VALID_PROGRAMS table.

5.3.10 New Users Visible Using OEM Console

If you add users of OCI Services to an Oracle Rdb release 7.1 or later database by using the add_user function of the RDB_NATCONNnn command procedure, they are now visible through the OEM console. A CREATE USER SQL command is executed by the add_user

function, which adds new users to the RDB\$PROFILE table. and makes them visible when you use the OEM console.

5.3.11 New Error Message for Unregistered Usernames

A new error message, ORA-1017, is returned at logon time if the username is not registered in the database. This message replaces the old message, ORA-1010. A message is put in the log file that says the username is not registered and that RDB_NATCOMM.COM should be run to add the username to the database. This message can be ignored if the application successfully connects.

5.3.12 Thin JDBC Access to Oracle Rdb Databases

Starting with SQL*Net for Rdb release 7.1.5, access to Oracle Rdb databases are supported through thin JDBC.

5.3.13 New Command File RDB_NATCONN.COM

There is a new command file called RDB_NATCONN.COM that 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:
Password Verification:
Username: username2
Password:
Passwprd Verification:
Username: <Enter>
```

Refer to the Oracle SQL/Services Server Configuration Guide Release 7.1.6 for more information.

5.3.14 New Logical Names

Three new logical names have been added to SQL*Net for Rdb:

- **SQLNET_DEBUG_FLAGS**

Support for debug flags has been added to SQL*Net for Rdb to enable additional information to be logged in the SQL/Services executor log file. See [Section 5.3.15](#) for additional details.

- **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.

- **SQLNET_DOMAIN**

SQL*Net for Rdb assumes that the default domain name is .WORLD. You can define the logical, SQLNET_DOMAIN, to change the domain name. This is especially relevant for people accessing SQL*Net for Rdb using dblinks from an Oracle database. The logical can be defined in the SQL initialization file for the service.

5.3.15 Support for Debug Flags

Support for debug flags has been added to SQL*Net for Rdb. These flags enable additional information to be logged in the SQL/Services executor log file. To enable this feature, define the logical SQLNET_DEBUG_FLAGS in your SQL/Services service process initialization

file. The following debug flags are currently supported. Note that these debug flags are case sensitive.

- H
Causes routine header information to be logged whenever a routine is executed.
- T
Causes the current time to be logged whenever header information is logged. This flag is dependent on the H flag. If the H flag is not specified, time information will not be logged.

For example, if the following line is inserted into your process initialization file, header information will be logged:

```
$ DEFINE SQLNET_DEBUG_FLAGS "H"
```

Information such as the following will appear in the executor log file. This information is especially useful when reporting problems to Oracle. For example:

```
... Server entered ...
GTZI_SERVER START UP -> Wed Aug  8 13:34:33 2001
*** GTZGPRD ***
*** GTAGPRD ***
NLS initialization ...
* GTWILAN: Setting LANGUAGE *
Language is AMERICAN_AMERICA.US7ASCII
```

If the following line is inserted into your process initialization file, header information and the current time will be logged:

```
$ DEFINE SQLNET_DEBUG_FLAGS "HT"
```

Information such as the following will appear in the executor log file:

```
... Server entered ...
Current Time = Wed Aug  8 13:34:33 2001

GTZI_SERVER START UP -> Wed Aug  8 13:34:33 2001

Current Time = Wed Aug  8 13:34:33 2001

*** GTZGPRD ***
Current Time = Wed Aug  8 13:34:33 2001

*** GTAGPRD ***
Current Time = Wed Aug  8 13:34:33 2001
```

```
NLS initialization ...
Current Time = Wed Aug  8 13:34:34 2001

* GTWILAN: Setting LANGUAGE *
Current Time = Wed Aug  8 13:34:34 2001

Language is AMERICAN_AMERICA.US7ASCII
Current Time = Wed Aug  8 13:34:34 2001
```

The following logical definition will not cause any additional information to be logged since the T debug flag is dependent on the H debug flag.

```
$ DEFINE SQLNET_DEBUG_FLAGS "T"
```

5.3.16 Support for Hot Standby

SQL*Net for Rdb supports Hot Standby databases.

5.3.17 SQL*Net for Rdb Product

SQL*Net for Rdb supports Oracle Rdb release 7.0 and higher 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.

5.3.18 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.

5.3.19 Support for New ALTER SESSION SET CONSTRAINTS Syntax

OCI Services for Oracle Rdb release 7.1.6 supports the ALTER SESSION SET CONSTRAINTS syntax added in Oracle Rdb release 7.1.2. Refer to the Oracle Rdb documentation for details on this new feature.

5.3.20 Support for RDB\$NATCONN_CUPPnn Utility

Refer to the Oracle SQL/Services Server Configuration Guide for information on how to use the RDB_NATCONNnn command file to prepare or upgrade an Oracle Rdb database, drop OCI-associated tables and functions from an Oracle Rdb database, or to add, remove, modify, or show users with encrypted passwords in a prepared Oracle Rdb database.

5.4 Software Errors Fixed in Previous OCI Services for Oracle Rdb Releases

This section highlights software errors fixed in previous OCI Services for Oracle Rdb releases.

5.4.1 Invalid ROWID Messages

There were two cases where the ORA-01410 invalid ROWID error message occurred. The error occurred if you were updating data from a row selected in a form; when you clicked on the update button, the error message appeared. The error was also seen if you were running a procedure using Procedure Builder. In both cases, the declare cursor statement contained the clause "for update of" and the update statement contained the clause "where current of <cursor>". This problem has been fixed in release 7.1.6.

5.4.2 Unable to Connect to Multiple Databases Using a Universal Service

Various errors were displayed if you tried to connect to multiple databases in a universal service. Sometimes the Rdb error RDB-F-REQ_WRONG_DB was returned from the connect statement; sometimes it appeared that SQLPlus had connected, but references to the second database failed. This problem has been fixed in release 7.1.6.

5.4.3 Milliseconds Supported for **TIMESTAMP** and **INTERVAL** Data Types

Milliseconds are supported if structured date types are turned on. You must issue one of the following commands:

```
ALTER SESSION STRUCTURED_DATE_TYPES ON  
OR
```

```
DEFINE SQLNET_STRUCTURED_DATE_TYPES Y
```

You must also issue the command:

```
ALTER SESSION SET NLS_TIMESTAMP_FORMAT = "yyyy-mm-dd hh24:mi:ss.ff"
```

5.4.4 Error **ORA-03106** Returned Instead of Expected **ORA-01722** Error

When a character value was entered into a **NUMBER** field, error **ORA-03106** was displayed instead of the correct error, **ORA-01722**. This has been corrected in release 7.1.6.

5.4.5 Additional Error Messages in **FORMS**

Additional error messages made Rdb errors too long to fit in standard error windows in **FORMS**. This problem has been fixed for the **ORA-0001** (index field value already exists) and **ORA-02290** (check constraint violated) errors.

5.4.6 **TIMESTAMP** Data Type Caused Error

Use of the **TIMESTAMP** data type resulted in an unimplemented or unreasonable conversion requested error. This has been corrected in release 7.1.6.

5.4.7 Error Using **DESCRIBE** Command

The **DESCRIBE** command failed when it was used on a table that was associated with a database link. This has been corrected in release 7.1.6.

5.4.8 Error Storing Null Value

Storing a null value in a column with the **LIST OF BYTE VARYING** data type caused a protocol violation. This has been corrected in release 7.1.6.

5.4.9 DOUBLE PRECISION Column Converted Incorrectly

A DOUBLE PRECISION column in an Rdb table was converted to NUMBER(53) instead of FLOAT(53) when it was accessed through OCI Services for Oracle Rdb. This has been corrected in release 7.1.6.

5.4.10 JDBC Errors

The following JDBC errors have been fixed in release 7.1.6:

- Protocol violation - An error occurred when a null value stored in a column with the LIST OF BYTE VARYING datatype was retrieved via the ResultSet.getBinaryStream method.
- Invalid datatype in OCI call - An error occurred when an attempt was made to store a value into a timestamp column via the PreparedStatement.setTimestamp method or when an attempt was made to store a null into a timestamp column via the PreparedStatement.setNull method.

5.4.11 Incorrect Oracle Errors Returned to OCI Clients

In releases prior to 7.1.5.9.1, if users connecting to an OCI universal service got an error during logon, instead of an Oracle error code Oracle SQL*Net for Rdb returned an Rdb error code to the OCI client.

This led to the Oracle client displaying the Oracle error text corresponding to the Rdb error code, which bore no relation to the actual error condition. The errors most often displayed were the following.

```
ORA-01028: internal two task error
ORA-00001: unique constraint violated
```

This problem has been fixed in Oracle SQL/Services / SQL*Net for Rdb release 7.1.5.9.1. The OCI client now displays the correct Oracle error.

5.4.12 Rdb Errors Unintentionally Suppressed

In releases prior to 7.1.5.9.1, if users connecting to an OCI database service got an error during logon, a "Database not setup..." message was added to the error stack, causing the error reported by Rdb to be pushed off the error stack and not be displayed. The following is an example of the resulting errors reported to the user.

```
ERROR: ORA-01031: insufficient privileges
Database not setup correctly for Oracle SQL*Net for Rdb.
```

For details, look in Oracle SQL/Services executor log file: ...

This problem has been fixed in Oracle SQL/Services / SQL*Net for Rdb release 7.1.5.9.1.
The OCI client will now display the Rdb error.

5.4.13 RDB_NATCONNnn Command Procedure No Longer Turns On Verify

The RDB_NATCONNnn command procedure has been fixed so that it no longer turns on verify. ON CTL_Y and ON ERROR have been added to force the deletion of the adduser.sql script that is created.

5.4.14 Changes to RDB_NATCONN_PREPAREnn and RDB_NATCONN_UPGRADExx Scripts

There have been many changes to the RDB_NATCONN_PREPAREnn and RDB_NATCONN_UPGRADExx scripts, including changes to the number of parameters passed to ORA_LOGIN2 and V\$NLS_UPDATE. Because of these changes, you must run RDB_NATCONN_UPGRADExx before you can connect to your databases using SQL*Net for Rdb.

If you do not upgrade your database, you will see the following error when you try to connect:

```
ERROR:
ORA-09100: Message 9100 not found; No message file for product=NATCONN,
facility=GTW
%SQL-F-ARGCOUNT, Procedure ORA_LOGIN2 expected 10 parameters, was passed 11
```

5.4.15 Persona Nopriv Error Using SQLplus and Other OCI Clients

Non-privileged users could not connect from SQLplus even though they were granted database access. The connect would succeed when 'persona support is disabled' but fail when 'persona support is enabled'.

The following example shows the problem:

```
UAF> show joe_nobody
Username: JOE_NOBODY                               Owner:
Account:                                           UIC:   [424,7] ([JOE_NOBODY])
CLI:      DCL                                       Tables: DCLTABLES
Default:   DISK$DKD600:[JOE_NOBODY]
LGICMD:
Flags:
Primary days:   Mon Tue Wed Thu Fri
```

```

Secondary days:                Sat Sun
No access restrictions
@ Expiration:                  (none)   Pwdminimum: 6   Login Fails: 0
@ Pwdlifetime:                 (none)   Pwdchange: 23-APR-2002 07:39
@ Last Login: 21-MAY-2002 11:39 (interactive), (none)
(non-interactive)
Maxjobs:      0   Fillm:      100   Byt1m:      64000
Maxacctjobs:  0   Shrfillm:    0   Pbyt1m:     0
Maxdetach:    0   BIO1m:      150   JTquota:    4096
Prclm:        8   DIO1m:      60   WSdef:      2000
Prio:         4   AST1m:     250   WSquo:      4000
Queprio:      4   TQElm:     10   WSextent:  16384
CPU:          (none)   Enqlm:    2000   Pgflquo:   50000
Authorized Privileges:
    NETMBX      TMPMBX
Default Privileges:
    NETMBX      TMPMBX
Identifier                    Value              Attributes
JOE                           %X80010015
READ_ONLY                      %X80010016

```

```

SQL> show prote on database rdb$dbhandle
Protection on Alias RDB$DBHANDLE
  (IDENTIFIER=SQLNET4RDB,ACCESS=SELECT+INSERT+UPDATE+DELETE+SHOW+CREATE+ALTER+
  DROP+DBCTRL+OPERATOR+DBADM+SECURITY+DISTRIBTRAN)
  (IDENTIFIER=JOE,ACCESS=SELECT+UPDATE)
  (IDENTIFIER=READ_ONLY,ACCESS=SELECT)
  (IDENTIFIER=[*,*],ACCESS=NONE)

```

Error from SQLplus when connecting as joe_nobody (after the service has started successfully):

```
ERROR: ORA-01031: insufficient privileges
```

Error in executor log file:

```

Rdb operation..: EXECUTE IMMEDIATE - LOGIN2
Rdb error...(0): %RDB-E-NO_PRIV, privilege denied by database facility

```

Possible workarounds include giving the user more privileges or rights, or disabling persona 'security checking is external (persona support is disabled)'.

This problem was corrected in Release 7.1.5.6.

5.4.16 Connections to SQL*Net for Rdb Would Hang

Connections to SQL*Net for Rdb from Oracle version 8.1.7 would sometimes hang. This problem has been corrected.

5.4.17 DESCRIBE of an Object Reported Datatype UNKNOWN

A DESCRIBE of an object, especially a procedure or function, often reported datatype UNKNOWN. The correct datatype is now reported.

5.4.18 NLS_CHARACTERSET Ignored in ALTER SESSION Command

If an ALTER SESSION command included the symbol NLS_CHARACTERSET, the command reported success but did not change the character set. The value of NLS_CHARACTERSET is determined at connection time and cannot be changed during the session. The command now returns the error ORA-12705.

5.4.19 Memory Leaks

Some memory leaks were fixed.

5.4.20 Fatal Error When Running Queries Longer Than 16K

If you ran queries that were longer than 16 K through SQL*Plus and SQL*Net, the process was truncated, and a fatal error was generated:

```
ERROR at line 1:  ORA-09100: Message 9100 not found; No message file
for product=NATCONN, facility=GTW
%SQL-F-UNTERM_C_STR ,Missing null terminator for C string
```

This problem has been fixed for release 7.1.5.6.

5.4.21 Error Accessing Rdb Database Using Dblink

In previous releases, if you used Oracle 9.0.1 and accessed an Rdb database using a dblink, you could see the following error:

```
ORA-00600: internal error code, arguments: [qctstc2o1], [1], [0],
[31],[96],
[2], [2], []
```

The following two SQL statements are the types of statements that would see the error:

```
insert into ora_table (select * from rdb_table@rdb.world);
```

```
create table ora_table as select col1 from rdb_table@rdb.world;
```

5.4.22 Running an Rdb Stored Procedure Using a Database Link

In previous versions, when you ran an Rdb stored procedure using a database link, it was possible to get the following error:

```
SQL> execute insert_col@rdb('abcde');
BEGIN insert_col@rdb('abcde'); END;
*
ERROR at line 1:
ORA-01401: inserted value too large for column
ORA-06512: at line 1
```

In the course of fixing this error, some changes were made to the parameter handling and conversion implementation. These changes can sometimes cause the following error to appear when the stored procedure is first run:

```
SQL> execute insert_col@rdb('abcde');
BEGIN insert_col@rdb('abcde'); END;
*
ERROR at line 1:
ORA-04068: existing state of packages has been discarded
ORA-04062: %s of %s has been changed
ORA-06512: at line 1
```

This error can be ignored; the procedure will run correctly when run again.

5.4.23 Protocol Mismatch Error

In previous versions, you could see a protocol mismatch error when you did a select from a table that had an index, as shown in the following example:

```
SQL> select count(*) from test_optim@rdb;
select count(*) from test_optim@rdb
*
ERROR at line 1:
ORA-02072: distributed database network protocol mismatch
```

5.4.24 SQL*PLUS Hangs Using SQL*Net for Rdb

If you had run the RDB_NATCONN_DBMSAPPL_PREPARE or the RDB_NATCONN_DBMSOUTPUT_PREPARE script on your Rdb database, SQL*Plus Version 9.0.1.0.1 would hang trying to connect using SQL*Net for Rdb. This no longer occurs.

5.4.25 Thick JDBC Driver Returned Error

In previous versions it was possible that the thick JDBC driver would return an error on the second of two select statements:

```
***** ACCESS BAD *****
java.sql.SQLException: ORA-01010: Ungultige (=invalid)
OCI-Operation
    at
oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:168)
    at
oracle.jdbc.oci8.OCIDBAccess.check_error(OCIDBAccess.java:1597)
    at
oracle.jdbc.oci8.OCIDBAccess.parseExecuteDescribe(OCIDBAccess.java:798)
```

The SQL*Net for Rdb log would show the following:

```
Rows fetched(1) are: 1
AL7 close [opncrs=0 GTOD_ALL->allopt=2048 err=1403
Function: Cancel All
Oracle error: 1010 for function: Cancel All
Function: Cursor close all
```

5.4.26 Select of Column Returned Data in Wrong Format

In previous versions of SQL*Net for Rdb, if you were using Rdb V7.0xx, a select of a column using TO_CHAR and a column alias could display the data as hex instead of characters:

```
sql> select to_char(1) from dual;
      shows the value 1

sql> select to_char(1) col_alias from dual;
      shows the value 31
```

5.4.27 Two-phase Commits Using Both Rdb and Oracle Databases

When several users were doing two-phase commits using a dblink including both an Rdb database and an Oracle database, they could see locks on the ORA_COMM_TRANS table that caused users to be blocked.

5.4.28 Logicals Now Defined for ORA_NLS, ORA_NLS32, and ORA_NLS33

In earlier versions of SQL*Net for Rdb, users that wanted to define a language other than AMERICAN had to define the logical names ORA_NLS, ORA_NLS32, and ORA_NLS33 to point to message files and other NLS files.

SQLSRV_MON71.EXE now contains definitions for all the ORA_NLS nn logical names and users no longer need to define them.

5.4.29 Dblink Failures Fixed

The following dblink problems in SQL*Net for Rdb releases 7.1.5 and 7.1.5.1 have been fixed:

- Taking an update lock on a table through dblink caused the connection to abort.

The SQL statement that failed was:

```
SELECT * FROM tab@dblink WHERE col='xxx' FOR UPDATE OF col2
```

The error returned was:

```
ERROR at line 1:  
ORA-03113: end-of-file on communication channel
```

- Creating a view through a dblink caused the connection to abort.

The SQL statement that failed was:

```
CREATE VIEW vu AS SELECT * FROM tab
```

The error returned was:

```
ERROR at line 1:  
ORA-03113: end-of-file on communication channel
```

- SQL functions such as USER and SYSDATE did not work correctly through a dblink.

The SQL statement that failed was:

```
SELECT * FROM tab@dblink WHERE col=USER
```

The error returned was:

```
ORA-00911: invalid character
%SQL-F-BAD-TOKEN, is not a valid SQL language element
```

5.4.30 Better Performance of SQL*Net for Rdb Startup

The startup time for a SQL*Net for Rdb session was very slow, especially for databases with many tables. For example, the following query was slow:

```
SELECT COUNT(*) from ORA_OBJECTS where NAME not in (SELECT TRIM (RDB$RELATION_
NAME from RDB$RELATIONS) and OTYPE = 1;
```

This query was intended to help keep the table, ORA_OBJECTS, synchronized with the tables that were actually in the database. The stored procedure, ORA_DELETE_PHANTOMS, was called if the count in the query was greater than 0. Since the query was taking so much time, and almost always returning 0, it was decided to remove the query from the startup and remove the call to the stored procedure, ORA_DELETE_PHANTOMS. It is now recommended that the stored procedure, ORA_DELETE_PHANTOMS, be run periodically, especially if tables are being created or dropped from both SQL and SQL*Plus. The following example shows how to invoke the stored procedure:

```
SQLPLUS> begin ora_delete_phantoms(); end;
SQLPLUS> /
```

If the application did not require strict schema emulation, the workaround was to add the following line to the SQL initialization file for the service:

```
ALTER SESSION SET SCHEMA EMULATION RELAXED;
```

This would disable the schema emulation mechanism.

5.4.31 Change Maximum CHAR and VARCHAR Sizes

For Oracle8, the maximum length of CHAR and VARCHAR datatypes changed from what the maximum was for Oracle7. Maximum character length changed from 255 to 2000; maximum VARCHAR length changed from 2000 to 4000. This affects the datatype mappings for SQL*Net for Rdb, since CHAR and VARCHAR strings larger than the Oracle maximum are mapped to long or blob.

5.4.32 Fixed Metadata Retrieval Using JDBC

Using JDBC, the metadata query, meta.getColumns, did not return data about any of the tables in the database except for GLOBAL_NAME.

The following example shows the JDBC code for the metadata query that was failing. Running this query for any table, other than GLOBAL_NAME, would produce no results. Following the code fragment is part fo the result set for the ALL_TABLES table.

```
DatabaseMetaData meta = conn.getMetaData();
ResultSet c = null;
c = meta.getColumns(null, null, args[0], "%");
while (c.next()) {

    System.out.println("\t" + c.getString(1) + ", " +
        c.getString(2) + ", " + c.getString(3) + ", " +
        c.getString(4) + ", " + c.getString(5) + ", " +
        c.getString(6));
    }
c.close();
```

Result Set:

```
null, SYS, ALL_TABLES, OWNER, 12, VARCHAR2
null, SYS, ALL_TABLES, TABLE_NAME, 12, VARCHAR2
null, SYS, ALL_TABLES, TABLESPACE_NAME, 12, VARCHAR2
null, SYS, ALL_TABLES, CLUSTER_NAME, 12, VARCHAR2
null, SYS, ALL_TABLES, IOT_NAME, 12, VARCHAR2
...
```

5.4.33 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.

5.4.34 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.

5.4.35 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;`

5.4.36 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
```

5.4.37 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.

5.4.38 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.

5.4.39 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.

5.4.40 Insufficient Memory or Quota Problem

In V7.1.5.6. of SQL*Net for Rdb, there was a problem during cleanup, when a user disconnected, that prevented resources from being properly released. This problem could manifest itself as insufficient memory, or insufficient BYTLM or FILLM quotas.

5.4.41 Problem Storing and Retrieving Long Raw Data

Using the thin JDBC driver with Release 7.1.6 of SQL*Net for Rdb, there was a problem in both storing and retrieving long raw (image) data. These problems have been fixed in , with the following restrictions:

- Image data must be defined as BinaryStream data, and should be stored using setBinaryStream and retrieved using getBinaryStream.
- For the thin JDBC driver, use of the 'blob' datatype is not yet supported. The logical SQLNET_BLOB must not be defined or must be defined as "N".
- The default maximum size of LONG RAW data is 100,000 bytes. If your long raw data is longer or significantly shorter than that, you should define the new logical SQLNET_MAXLONGRAW to the value of your longest long raw data. Long raw data must be stored in a single insert statement and retrieved in a single get statement. The SQLNET_MAXLONGRAW value is used to allocate the buffer to hold the data, so it must be large enough to hold the entire value of the image data.

5.4.42 Problem Retrieving Non-English Long Raw Data

In previous releases, when using a language other than English, data defined as LONG RAW was not being fetched correctly. Image data was being converted as if it were character data in the case where NLS_LANG was defined to be other than English.

5.4.43 Problem Retrieving Long Raw Data Shorter Than 255 Bytes

In previous releases, image or LONG RAW data that was shorter than 255 bytes was not being fetched and returned correctly. Following is an example of a JAVA return showing the error:

```
D:\TAR\1141042\testThin901\TESTCASE>java TestLongRawRead
java.sql.SQLException: Io exception: Protocol violation at
oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:180) at
oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:222) at
oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:335) at
oracle.jdbc.dbaccess.DBDataSetImpl.getItem(DBDataSetImpl.
java:1157)
```

5.4.44 Problem Starting Read/Write Transactions

With previous releases, users were not able to start a read/write transaction when using a character set other than DEC_MCS. Following is the error the user would see:

```
SQL> insert into tabl values (1);
insert into tabl values (1)
*
ERROR at line 1:
ORA-01456: may not perform insert/delete/update operation inside a
READ ONLY transaction
```

The executor log file shows the error:

```
gtschema:: schema: <NULL>, object: ORA_OBJECTS, action: REF_TABLE
Rdb operation.: EXECUTE IMMEDIATE - CHECK_READ_ONLY
Rdb error...(0): %SQL-F-INCCSCMP, Incompatible character set
comparison between NAME and <value expression
```

5.4.45 Change to SQL*Net for Rdb Logging

In previous releases, SQL*Net for Rdb turned on logging during user connection and logon. Starting with Release 7.1.6, SQL*Net for Rdb will no longer turn on logging, so executor log files will not contain the same information as in prior versions. The behavior will be the same as prior versions if users turn on logging.

If logging is turned on and a SET SESSION AUTHORIZATION statement is being logged, as it always is for universal services, only "SET SESSION AUTHORIZATION" will be logged, not the rest of the statement which contains the username and password.

5.4.46 Updated RDB_NATCONN_xxx_PREPARE.SQL Script

New versions of the two optional SQL scripts RDB_NATCONN_DBMSAPPL_PREPARE and RDB_NATCONN_DBMSOUTPUT_PREPARE are provided with the installation of SQL*Net for Rdb release 7.1.5.7. The updated scripts fix problems that occurred when running the scripts attached to a DEC_KANJI database. In previous releases, SQL*Net for

Rdb users would see the incompatible character set message, INCCSASS. Executing prior versions of these scripts against databases with character sets other than DEC_MCS may also experience the same error. If so, please run the revised scripts provided by release 7.1.5.7.

5.4.47 Displayed Release Type Corrected

With releases prior to , Oracle SQL*Net for Rdb always displayed "Development" as the release type for Oracle Rdb OCI Server, regardless of the type of kit installed. This could be seen in the SQLPLUS welcome banner or by invoking the "SELECT * FROM V\$VERSION" command.

This problem has been fixed in Oracle SQL*Net for Rdb . "Production" will now be displayed for production releases.

5.4.48 Detection of Repeated Intrusion Attempts.

Code has been added to SQL*Net for Rdb to detect repeated intrusion attempts. This code is activated by a user whose username and/or password are not in the USERS\$ table of the Rdb database. Such a user may be locked out of the system, even if he has a valid OpenVMS account, if he exceeds the parameters set at the site for intrusion detection and denial.

5.4.49 Changes to Scripts

The scripts were changed to fix the problem where after a user had been deleted from the database, other users could not be added.

Changes were also made in the scripts to fix problems of retrieving table and column names in some circumstances.

5.4.50 Using Two-Phase Commit from a dblink

Users wanting to use two-phase commit from a dblink in an Oracle database to a single Rdb database must execute the following command:

```
alter session set tx_mode nowarn_lpc
```

It can be included in the sql init file for the service or it can be executed from the client. This instructs SQL*Net for Rdb that it should join the distributed transaction and allow the Oracle database to co-ordinate the transaction. If this is not done, SQL*Net for Rdb will return an error saying it can't join the distributed transaction.

For example:

```
$sqlplus username/password
```

```
SQL*Plus: Release 8.0.5.0.2 - Production on Tue Sep 23 14:13:30 2003

(c) Copyright 1998 Oracle Corporation. All rights reserved.

Connected to:
Oracle8 Enterprise Edition Release 8.0.5.1.0 - Production PL/SQL Release
8.0.5.1.0 - Production
SQL> insert into t1 values (1,'abc');

1 row created.

SQL>
insert into t1@dblink1 values (1,'abc');
insert into t1@dblink1 values (1,'abc')
*
ERROR at line 1:
ORA-02047: cannot join the distributed transaction in progress
```

5.4.51 Determining the State of a Transaction When a Two-Phase Commit Fails

If there is a failure in a two-phase commit transaction, the Oracle database that is coordinating the transaction does a new connect to the Rdb database to determine the state of the transaction. In order to do this connection, there needs to be a username and password that SQL*Net for Rdb can use to connect to the Rdb database and retrieve the information from the ORA_COMM_TRANS table. The logical SQLNET_RECO_USER must be defined either as a system logical or in the process initialization file of the service. The format of the logical is as follows:

```
$ define SQLNET_RECO_USER "/" "USERNAME/PASSWORD"
```

5.4.52 Prestarted Transactions Left Open After a Commit Is Issued

Performing a two-phase commit transaction with an RDBMS database and an Rdb database always leaves a pre-started transaction open on the Rdb database after a commit is issued. This potentially stalls an online rmu/backup unless /noquiet is specified. This has been fixed by a change to the ORA_TRANS module which will be executed by running the script RDB_NATCONN_UPGRADE[71].SQL. The module ORA_TRANS will be replaced with a fixed version.

5.4.53 Using a dblink to a Service Where NLS_LANG Is Defined As Other Than the Default

A change was made to fix the problem of using a dblink to a service where NLS_LANG is defined as other than the default.

Example:

```
% sqlplus scott/tiger
```

```
SQL*Plus: Release 8.1.7.0.0 - Production on Thu Apr 10 15:32:11 2003
@ (c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
Connected to:
```

```
Oracle8i Enterprise Edition Release 8.1.7.2.0 - Production
```

```
With the Partitioning option
```

```
JServer Release 8.1.7.2.0 - Production
```

```
SQL> select * from v$version@rdb71;
```

```
select * from v$version@rdb71
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02068: following severe error from RDB71
```

```
ORA-03113: end-of-file on communication channel
```

```
Excerpt from the log file:
```

```
Client data for fetch item 4
```

```
Client data: column_name <NAME>
```

```
Client data: type 1, length 17, nul/ind 0, precision 0, scale 0 Client text
```

```
<SPECIAL FUNCTIONS> Rows fetched(1) are: 2
```

```
Function: CLOSE
```

```
gta: gtaclse: release3(1)
```

```
Function: OK2RPC
```

```
statement.....(1): begin call v$nls_set_func(); end
```

```
Prescribe.
```

```
gta: gtaclse: release3(1)
```

```
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual address=00000000
```

```
0000464, PC=00000000003EF0C0, PS=0000001B
```

```
Improperly handled condition, image exit forced.
```

```
Signal arguments: Number = 0000000000000005
```

```
Name = 000000000000000C
```

```
0000000000000000
```

```
0000000000000000
```

```
0000000000000464
```

```
00000000003EF0C0
```

000000000000001B

5.4.54 Misleading Logging Removed

Some logging was removed from error logging which was misleading about the error code and the cause of the error. Other logging was removed as not useful.

5.4.55 Schemas Define for Users in the USER\$ Table

All users in the USER\$ table, as well as all users who are creators of tables, now have schemas defined for them.

5.4.56 Using a Bind Variable Twice in a SQL Statement

There was a problem when using a bind variable twice in a SQL statement; that problem has been fixed.

Following is an example of a query that would cause the error:

```
SQL> select sh.employee_id, sh.salary_amount
  2  from salary_history sh
  3  where sh.employee_id = :p_employee_id
  4  and sh.salary_amount = (select max (sh2.salary_amount) from salary_history
sh2 where sh2.employee_id = :p_employee_id); select sh.employee_id, sh.salary_
amount
*
```

ERROR at line 1:
ORA-03115: unsupported network datatype or representation

5.4.57 Testing for a Read-only Database

A change was made in the login code to test for a read-only database in a different way that does not cause possible priv errors. These errors would be seen in a log file, although they didn't cause the connection to fail. This priv error at check_read_only will no longer happen.

5.4.58 FILLM Quota Problem

When using a language other than the default (American), fillm quota would be used up by disconnecting and connecting to a service.

5.4.59 NLS_DATABASE_PARAMETERS Added to Scripts

The view NLS_DATABASE_PARAMETERS has been added to the scripts, as well as the parameter NLS_NCHAR_CHARACTERSET. These are both required by ODP.NET.

5.4.60 RDB_NATCONN.COM Problem Fixed

There was a problem when the command file RDB_NATCONN.COM was used in an environment where a sql initialization file (sqlini) was defined and contained an attach statement. Sqlini is now defined to be nl: in the command file.

Example:

```
SQLUSER71> define sql$database mf_personnel
SQLUSER71> define sqlini "dkd100:[sqluser71]sql.ini"
(NOTE: sqlini file contains attach 'filename sql$database';)
SQLUSER71> @sys$library:rdb_natconn
  Operation (prepare/upgrade/drop/adduser): adduser
  Database: sql$database
  Username: sqluser70
@   Password:
@   Password Verification:
  Username:
  This alias has already been declared.
  %SQL-F-DEFDBDEC, A database has already been declared with the default
alias
  %RDB-E-EXTFUN_FAIL, external routine failed to compile or execute successfully
@   -RDMS-E-RTN_ERROR, routine "ORA_ENCRYPT_PASSWORD" generated an
error
during
  execution
  -SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
address=000000000000
0010, PC=0000000019B3DB8, PS=0000001B
```

5.4.61 Problem with Universal Service Fixed

When using a universal service, the username would be set to the service owner rather than the connecting user.

This was fixed for release 7.1.5.8.

5.4.62 Memory Leak Related to dblink Fixed

There was a memory leak when doing many inserts using a dblink. This was fixed for release 7.1.5.8.

5.4.63 Connection Problem Fixed

If an error occurred when a user logged on, the next user who tried to connect would get the following error:

```
Assertion failed: "find_mblock(current_mgroup,mblock) == current_mgroup"  
in file NATCONN$SRC_V071573:[CODE]GTME.C;1 at line 468.
```

This bug was fixed in release 7.1.5.8.

5.4.64 Read Only Transactions Started During Connection

There was a problem in which SQL*Net for Rdb would start a READ ONLY transaction during connection. this caused problems for databases running with snapshots deferred. This problem has been fixed so that SQL*Net for Rdb now starts a transaction in an appropriate mode, as determined by Rdb.

5.4.65 SQL*Plus Invocation Failed with some NLS_LANG Definitions

Some versions of SQL*Plus, when invoked with NLS_LANG defined to be something other than the default (AMERICAN_AMERICA), would cause later connection attempts by other client tools to fail. For example, the following error would be returned to a JAVA program:

```
Exception thrown java.sql.SQLException: ORA-00900: invalid SQL statement ,8  
E85wwE_F E zE85
```

The log might also show a result similar to the following:

```
>>>> NEW SESSION USER: scott PROGRAM: JDBC AT tue oct 07 17:14:37  
<<<<<<  
>>>> GTA compiled at Apr 25 2003 10:52:43  
gtschema:: schema: <NULL>, object: ORA_OBJECTS, action: REF_TABLE  
statement.....: 0"D?
```

This problem was fixed in release 7.1.5.8.