

Oracle® Transparent Gateway for DRDA

Installation and User's Guide

10g Release 1 (10.1) for UNIX

Part No. B12009-01

January 2004

Oracle Transparent Gateway for DRDA Installation and User's Guide, 10g Release 1 (10.1) for UNIX

Part No. B12009-01

Copyright © 2001, 2004, Oracle. All rights reserved.

Primary Author: Platform Technologies Division

The Programs (which include both the software and documentation) contain proprietary information; 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, except to the extent required to obtain interoperability with other independently created software or as specified by law, 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. This document is not warranted to be 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.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States 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 we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. 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	xvii
Preface.....	xix
Intended Audience	xix
Processors.....	xx
Documentation Accessibility	xx
Related Documents.....	xxi
Conventions.....	xxi
SQL*Plus Prompts	xxii
Storage Measurements	xxii
 1 Introduction	
1.1 Introduction to the Oracle Transparent Gateway	1-2
1.1.1 Protection of Current Investment	1-2
1.2 Release 10g Gateways	1-2
1.2.1 Advantages of the Gateway	1-3
1.3 Gateway Capabilities	1-3
1.3.1 Transparency at All Levels	1-3
1.3.2 Extended Database Services	1-4
1.3.3 Extended Advanced Networking, Internet and Intranet Support.....	1-4
1.3.4 Dynamic Dictionary Mapping	1-5
1.3.5 SQL	1-5
1.3.6 Data Definition Language	1-5
1.3.7 Data Control Language	1-5

1.3.8	Passthrough and Native DB2 SQL	1-5
1.3.9	Stored Procedures	1-5
1.3.10	Languages.....	1-6
1.3.11	Oracle Database server Technology and Tools	1-6
1.3.12	SQL*Plus	1-6
1.3.13	Two-Phase Commit and Multi-site Transactions	1-6
1.3.14	Site Autonomy	1-7
1.3.15	Migration and Coexistence	1-7
1.3.16	Security	1-7
1.4	Terms.....	1-7
1.5	Architecture	1-8
1.6	Implementation.....	1-9
1.7	How the Gateway Works	1-10
1.7.1	SQL Differences	1-10
1.8	Oracle Tools and the Gateway.....	1-10
1.8.1	SQL*Plus	1-11
1.9	Features	1-11
1.9.1	Heterogeneous Services Architecture	1-11
1.9.2	Performance Enhancements	1-11
1.9.3	Fetch Reblocking	1-11
1.9.4	Oracle Database 10g Passthrough Supported	1-11
1.9.5	Retrieving Result Sets Through Passthrough	1-11
1.9.6	Support for TCP/IP	1-12
1.9.7	Native Semantics	1-12
1.9.8	Columns Supported in a Result Set	1-12
1.9.9	EXPLAIN_PLAN Improvement.....	1-12
1.9.10	Heterogeneous Database Integration	1-12
1.9.11	Minimum Impact on Existing Systems	1-12
1.9.12	Large Base of Data Access.....	1-12
1.9.13	Application Portability	1-12
1.9.14	Remote Data Access	1-13
1.9.15	Support for Distributed Applications	1-13
1.9.16	Application Development and End User Tools.....	1-14
1.9.17	Password Encryption Utility	1-14
1.9.18	Support for DB2/OS390 V6 and V7 Stored Procedures	1-14

1.9.19	Codepage Map Facility.....	1-14
1.9.20	IBM DB2 Universal Database Support.....	1-14
1.9.21	IBM DB2 Version 5.1 ASCII Tables.....	1-14
1.9.22	Read-Only Support	1-14

2 Release Information

2.1	Product Set	2-2
2.2	Changes and Enhancements	2-2
2.3	Bugs Fixed in Release 10.....	2-2
2.4	Known Problems	2-3
2.5	Known Restrictions	2-4
2.5.1	DB2 Considerations	2-4
2.5.2	SQL Limitations.....	2-6

3 System Requirements

3.1	Hardware Requirements	3-2
3.1.1	Processors	3-2
3.1.2	Memory	3-2
3.1.3	Network Attachment.....	3-2
3.1.4	CD-ROM Drive.....	3-2
3.1.5	Disk Space	3-3
3.2	Software Requirements	3-3
3.2.1	Operating System.....	3-3
3.2.2	DRDA Databases	3-3
3.2.3	Communication Server	3-3
3.2.4	Oracle Database server	3-4
3.2.5	Oracle Networking Products.....	3-4
3.3	Documentation Requirements.....	3-4

4 Installing the Gateway

4.1	Introduction.....	4-2
4.2	Before You Begin	4-2
4.3	Checklist for Gateway Installation.....	4-3
4.4	Installation Overview	4-4

4.5	Before Beginning Installation.....	4-4
4.6	Installing the Gateway from CD-ROM	4-4
4.6.1	Step 1: Log on to the host	4-4
4.6.2	Step 2: Create the product installation directory	4-4
4.6.3	Step 3: Set the ORACLE_HOME environment variable.....	4-4
4.6.4	Step 4: Mount the CD-ROM.....	4-5
4.6.5	Step 5: Set the DISPLAY Variable	4-5
4.6.6	Step 6: Start the Oracle Universal Installer	4-6
4.6.7	Step 7: Step through the Oracle Universal Installer	4-6
4.6.8	Step 8: Verify installation success	4-6
4.7	Installation Complete.....	4-7
4.7.1	De-installing the Gateway.....	4-7

5 Configuring the DRDA Server

5.1	Checklists for Configuring the DRDA Server	5-2
5.1.1	DB2/OS390.....	5-2
5.1.2	DB2/400.....	5-2
5.1.3	DB2/UDB (Universal Database)	5-2
5.1.4	DB2/VM	5-2
5.2	DB2/OS390.....	5-3
5.2.1	Step 1: Configure the Communications Server	5-3
5.2.2	Step 2: Define the user ID that owns the package	5-3
5.2.3	Step 3: Define the recovery user ID	5-3
5.2.4	Step 4: Determine DRDA location name for DB2 instance	5-4
5.2.5	Step 5: Configure DB2 Distributed Data Facility for gateway.....	5-4
5.3	DB2/400	5-4
5.3.1	Step 1: Configure the Communications Server	5-4
5.3.2	Step 2: Define the user ID that owns the package	5-4
5.3.3	Step 3: Define the recovery user ID	5-5
5.3.4	Step 4: Determine DRDA location name for DB2/400 instance	5-5
5.4	DB2/UDB (Universal Database)	5-5
5.4.1	Step 1: Configure the SNA Communications Server	5-5
5.4.2	Step 2: Define the user ID that owns the package	5-6
5.4.3	Step 3: Define the recovery user ID	5-6
5.4.4	Step 4: Determine DRDA location name for DB2/UDB instance	5-6

5.5	DB2/VM	5-7
5.5.1	Step 1: Configure the Communications Server.....	5-7
5.5.2	Step 2: Define the user ID that owns the package	5-7
5.5.3	Step 3: Define the recovery user ID	5-7
5.5.4	Step 4: Determine DRDA location name for DB2/VM instance	5-7

6 Configuring SunLink for the Gateway

6.1	Checklist for Configuring the Communications Interfaces	6-2
6.2	Step 1: Setting up a Gateway Name	6-3
6.3	Step 2: Setting up a Configuration File	6-3
6.3.1	Starting the SunLink Peer-to-Peer Version 9 Software	6-4
6.4	Step 3: Side Information File.....	6-4
6.4.1	Partner_LU_name	6-4
6.4.2	Mode_name.....	6-4
6.4.3	TP_name	6-4
6.4.4	Sample Side Information File for Version 9	6-5
6.5	Step 4: Test the Connection.....	6-5
6.6	Using SNA Session Security Validation	6-5
6.7	SNA Conversation Security	6-5
6.7.1	SNA Security Option SECURITY=PROGRAM	6-5
6.7.2	SNA Security Option SECURITY=SAME.....	6-6

7 Configuring SNAP-IX Interfaces

7.1	Steps for Configuring the Communications Interfaces	7-2
7.2	Before You Begin	7-2
7.3	SNAP-IX Configuration Tool.....	7-2
7.4	Creating SNAP-IX Profiles for the Gateway	7-2
7.5	Independent Versus Dependent LUs	7-2
7.6	Creating SNA Definitions for the Gateway	7-3
7.6.1	Sample SNAP-IX Definitions.....	7-3
7.6.2	Configuring SNAP-IX.....	7-3
7.6.3	Invoking xsnaadmin	7-3
7.7	Using SNA Session Security Validation	7-13
7.8	SNA Conversation Security	7-13
7.8.1	SNA Security Option SECURITY=PROGRAM	7-13

7.8.2	SNA Security Option SECURITY=SAME	7-14
7.9	Testing the Connection	7-15

8 Configuring IBM Communication Server

8.1	Checklist for Configuring the Communications Interfaces	8-2
8.2	Step 1: Configuring Communication Server Profiles	8-2
8.3	Step 2: Creating Communication Server Profiles for the Gateway	8-2
8.3.1	Sample Profile Definitions	8-2
8.3.2	Profile Types	8-2
8.4	Step 3: Testing the Connection	8-5
8.5	Using SNA Session Security Validation	8-5
8.6	SNA Conversation Security	8-5
8.6.1	SNA Security Option SECURITY=PROGRAM	8-5
8.6.2	SNA Security Option SECURITY=SAME	8-6

9 Configuring SNAPPlus2

9.1	Steps for Configuring the Communications Interfaces.....	9-2
9.2	Before You Begin	9-2
9.3	SNAPPlus2 Configuration Tool.....	9-2
9.4	Creating SNAPPlus2 Profiles for the Gateway.....	9-2
9.5	Independent Versus Dependent LUs	9-2
9.6	Creating SNA Definitions for the Gateway	9-3
9.6.1	Sample SNAPPlus2 Definitions.....	9-3
9.6.2	Configuring SNAPPlus2.....	9-3
9.6.3	Invoking xsnapadmin.....	9-4
9.7	Using SNA Session Security Validation	9-13
9.8	SNA Conversation Security	9-13
9.8.1	SNA Security Option SECURITY=PROGRAM	9-13
9.8.2	SNA Security Option SECURITY=SAME.....	9-14
9.9	Testing the Connection	9-14

10 Configuring TCP/IP

10.1	Before You Begin	10-2
10.2	Configuring TCP/IP under UNIX	10-2

11 Oracle Net

11.1	Checklists for Oracle Net.....	11-2
11.1.1	Configuring Oracle Net.....	11-2
11.1.2	Advanced Security Encryption	11-2
11.2	Oracle Net Introduction	11-3
11.3	Oracle Net Overview	11-3
11.3.1	Distributed Processing	11-3
11.3.2	Distributed Database	11-3
11.3.3	Terminology for Oracle Net	11-4
11.4	Configuring Oracle Net.....	11-4
11.4.1	Step 1: Modify listener.ora file	11-4
11.4.2	Step 2: Modify tnsnames.ora file.....	11-5
11.5	Advanced Security Encryption	11-5
11.6	Setting Up Advanced Security Encryption for Test	11-6
11.6.1	Step 1: Set Advanced Security Encryption Parameters for the Gateway	11-6
11.6.2	Step 2: Set Advanced Security Encryption Parameters for Oracle Server	11-6
11.7	Testing Advanced Security Encryptions	11-6
11.7.1	Step 1: Connect Gateway and Oracle Integrating Server	11-6
11.7.2	Step 2: Reset Configuration Parameters on the Gateway.....	11-7

12 Configuring the Gateway

12.1	Configuration Checklists	12-2
12.2	Choosing a Gateway System Identifier (SID).....	12-4
12.3	Gateway Configuration	12-4
12.4	Configuring the Host	12-4
12.4.1	Step 1: Choose the <i>initsid.ora</i> file	12-4
12.4.2	Step 2: Tailor the initsid.ora file.....	12-5
12.4.3	Binding the DRDA Gateway Package.....	12-5
12.4.4	Binding Packages on DB2/Universal Database (DB2/UDB)	12-6
12.5	DRDA Gateway Package Considerations.....	12-7
12.5.1	Before Binding the DRDA Gateway Package	12-7
12.5.2	Sample SQL scripts	12-7
12.6	Backup and Recovery of Gateway Configuration	12-8
12.7	Configuring the Oracle Integrating Server	12-8
12.7.1	Step 1: Create a database link	12-8

12.7.2	Step 2: Create synonyms and views	12-9
12.8	Accessing the Gateway from Other Oracle Database Servers	12-9
12.9	Accessing Other DRDA Servers	12-9
12.10	Gateway Installation and Configuration Complete	12-9

13 Using the Gateway

13.1	Processing a Database Link.....	13-2
13.1.1	Creating Database Links	13-2
13.1.2	Guidelines for Database Links	13-2
13.1.3	Dropping Database Links	13-2
13.1.4	Examining Available Database Links.....	13-3
13.1.5	Limiting the Number of Active Database Links.....	13-3
13.2	Accessing the Gateway	13-3
13.2.1	Step 1: Login to the Oracle Integrating Server	13-3
13.2.2	Step 2: Create a database link to the DRDA database.....	13-3
13.2.3	Step 3: Retrieve data from the DRDA database.....	13-3
13.3	Accessing AS/400 File Members.....	13-3
13.4	Using the Synonym Feature.....	13-4
13.5	Performing Distributed Queries	13-4
13.5.1	Example of a Distributed Query	13-4
13.5.2	Two-Phase Commit Processing.....	13-5
13.5.3	Distributed DRDA Transactions	13-5
13.6	Read-Only Gateway	13-5
13.7	Replicating in a Heterogeneous Environment	13-6
13.7.1	Oracle Database 10g Server Triggers.....	13-6
13.7.2	Oracle Snapshots	13-6
13.8	Copying Data from Oracle Database 10g Server to DRDA Server.....	13-6
13.9	Copying Data from DRDA Server to Oracle Database 10g Server.....	13-7
13.10	Tracing SQL Statements	13-7

14 Developing Applications

14.1	Gateway Appearance to Application Programs	14-2
14.1.1	Fetch Reblocking	14-2
14.2	Using Oracle Stored Procedures with the Gateway	14-3
14.3	Using DRDA Server Stored Procedures with the Gateway	14-4

14.3.1	Oracle Application and DRDA Server Stored Procedure Completion	14-5
14.3.2	Procedural Feature Considerations with DB2	14-6
14.4	Database Link Behavior.....	14-6
14.5	Oracle Database Server SQL Construct Processing.....	14-6
14.5.1	Compatible SQL Functions.....	14-7
14.5.2	Translated SQL Functions.....	14-7
14.5.3	Compensated SQL Functions.....	14-7
14.5.4	Native Semantic SQL Functions	14-8
14.5.5	DB2/OS390 SQL Compatibility	14-8
14.5.6	DB2/Universal Database SQL Compatibility	14-10
14.5.7	DB2/400 SQL Compatibility	14-13
14.5.8	DB2/VM SQL Compatibility.....	14-16
14.6	Native Semantics	14-19
14.6.1	SQL Functions That Can Be Enabled	14-19
14.6.2	SQL Functions That Can Be Disabled	14-21
14.6.3	SQL Set Operators and Clauses	14-21
14.7	DRDA Datatype to Oracle Datatype Conversion	14-22
14.7.1	Performing Character String Operations	14-23
14.7.2	Converting Character String Datatypes	14-23
14.7.3	Performing Graphic String Operations.....	14-23
14.7.4	Performing Date and Time Operations	14-24
14.7.5	Dates.....	14-25
14.7.6	HS_NLS_DATE_FORMAT Support	14-26
14.7.7	Oracle TO_DATE Function.....	14-26
14.7.8	Performing Numeric Datatype Operations	14-27
14.7.9	Mapping the COUNT Function	14-27
14.7.10	Performing Zoned Decimal Operations.....	14-27
14.8	Passing Native SQL Statements through the Gateway.....	14-28
14.8.1	Using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE	14-28
14.8.2	Retrieving Results Sets Through Passthrough.....	14-29
14.9	Oracle Data Dictionary Emulation on a DRDA Server.....	14-30
14.9.1	Using the Gateway Data Dictionary.....	14-30
14.9.2	Using the DRDA Catalog.....	14-30
14.10	Defining the Number of DRDA Cursors	14-30

15 Security Considerations

15.1	Security Overview	15-2
15.2	Authenticating Application Logons	15-2
15.3	Defining and Controlling Database Links	15-3
15.3.1	Link Accessibility	15-3
15.3.2	Links and CONNECT Clauses	15-3
15.4	TCP/IP Security	15-3
15.5	Processing Inbound Connections.....	15-3
15.5.1	User ID Mapping.....	15-3
15.6	Passwords in the Gateway Initialization File	15-5
15.7	Using the g4drpwd Utility	15-6

16 Migration and Coexistence with Existing Gateways

16.1	Migrating Existing V4, V8, or V9 Gateway Instances to New Release	16-2
16.1.1	Step 1: Install the new Release	16-2
16.1.2	Step 2: Transferring initsid.gtwboot Gateway Boot Initialization parameters.	16-2
16.1.3	Step 3: Transferring initsid.ora Gateway Initialization File parameters.	16-2
16.2	Backout Considerations When Migrating to New Releases.....	16-2
16.3	New and Changed Parameters When Migrating to Release 10.....	16-3
16.3.1	New Parameters	16-3
16.3.2	Parameters That Have Been Changed in Usage	16-3
16.3.3	Parameters That Have Been Renamed	16-3
16.3.4	Obsolete Parameters	16-4
16.4	DRDA Server Considerations	16-4
16.5	Oracle Net Considerations	16-4

17 Error Messages, Diagnosis, and Reporting

17.1	Interpreting Gateway Error Messages.....	17-2
17.1.1	Errors Detected by the Oracle Integrating Server.....	17-2
17.1.2	Errors Detected by the Gateway	17-2
17.1.3	Errors Detected in the DRDA Software	17-3
17.1.4	Communication Errors	17-3
17.1.5	Errors Detected by the Server Database.....	17-3
17.2	Mapped Errors	17-4

17.3	Gateway Error Codes.....	17-5
17.4	SQL Tracing and the Gateway	17-6
17.4.1	SQL Tracing in the Oracle Database.....	17-6
17.4.2	SQL Tracing in the Gateway.....	17-7

A Oracle DB2 Data Dictionary Views

A.1	Supported Views	A-2
A.2	Data Dictionary View Tables.....	A-3
A.2.1	ALL_CATALOG	A-3
A.2.2	ALL_COL_COMMENTS	A-3
A.2.3	ALL_CONS_COLUMNS	A-3
A.2.4	ALL_CONSTRAINTS	A-4
A.2.5	ALL_INDEXES	A-4
A.2.6	ALL_IND_COLUMNS	A-6
A.2.7	ALL_OBJECTS	A-6
A.2.8	ALL_SYNONYMS	A-7
A.2.9	ALL_TABLES	A-7
A.2.10	ALL_TAB_COLUMNS	A-9
A.2.11	ALL_TAB_COMMENTS	A-10
A.2.12	ALL_USERS	A-10
A.2.13	ALL_VIEWS	A-10
A.2.14	COLUMN_PRIVILEGES.....	A-11
A.2.15	DICTIONARY.....	A-11
A.2.16	DUAL.....	A-11
A.2.17	TABLE_PRIVILEGES	A-12
A.2.18	USER_CATALOG	A-12
A.2.19	USER_COL_COMMENTS	A-12
A.2.20	USER_CONSTRAINTS	A-13
A.2.21	USER_CONS_COLUMNS	A-13
A.2.22	USER_INDEXES	A-14
A.2.23	USER_OBJECTS	A-15
A.2.24	USER_SYNONYMS	A-16
A.2.25	USER_TABLES	A-16
A.2.26	USER_TAB_COLUMNS	A-18
A.2.27	USER_TAB_COMMENTS	A-19

A.2.28	USER_USERS	A-19
A.2.29	USER_VIEWS.....	A-19

B Sample Files

B.1	Sample Gateway Initialization File.....	B-2
B.2	Sample Oracle Net tnsnames.ora File.....	B-3
B.3	Sample Oracle Net listener.ora File.....	B-4

C DRDA-Specific Parameters

C.1	Modifying the Gateway Initialization File.....	C-2
C.2	Setting Parameters in the Gateway Initialization File.....	C-2
C.3	Syntax and Usage	C-2
C.4	Gateway Initialization File Parameters	C-2
C.4.1	DRDA_CACHE_TABLE_DESC	C-2
C.4.2	DRDA_CAPABILITY.....	C-3
C.4.3	DRDA_CODEPAGE_MAP	C-3
C.4.4	DRDA_COMM_BUFLN	C-3
C.4.5	DRDA_CONNECT_PARM (SNA format)	C-3
C.4.6	DRDA_CONNECT_PARM (TCP/IP format)	C-3
C.4.7	DRDA_CMSRC_CM_IMMEDIATE	C-4
C.4.8	DRDA_DEFAULT_CCSD.....	C-4
C.4.9	DRDA_DESCRIBE_TABLE.....	C-4
C.4.10	DRDA_DISABLE_CALL	C-5
C.4.11	DRDA_FLUSH_CACHE	C-5
C.4.12	DRDA_GRAPHIC_PAD_SIZE.....	C-5
C.4.13	DRDA_GRAPHIC_LIT_CHECK.....	C-6
C.4.14	DRDA_GRAPHIC_TO_MBCS	C-6
C.4.15	DRDA_ISOLATION_LEVEL.....	C-6
C.4.16	DRDA_LOCAL_NODE_NAME	C-7
C.4.17	DRDA_MBCS_TO_GRAPHIC	C-7
C.4.18	DRDA_OPTIMIZE_QUERY	C-7
C.4.19	DRDA_PACKAGE_COLLID.....	C-7
C.4.20	DRDA_PACKAGE_CONSTOKEN	C-8
C.4.21	DRDA_PACKAGE_NAME	C-8
C.4.22	DRDA_PACKAGE_OWNER	C-8

C.4.23	DRDA_PACKAGE_SECTIONS	C-9
C.4.24	DRDA_READ_ONLY	C-9
C.4.25	DRDA_RECOVERY_PASSWORD	C-9
C.4.26	DRDA_RECOVERY_USERID	C-9
C.4.27	DRDA_REMOTE_DB_NAME.....	C-10
C.4.28	DRDA_SECURITY_TYPE	C-10
C.4.29	FDS_CLASS.....	C-10
C.4.30	FDS_CLASS_VERSION.....	C-10
C.4.31	FDS_INSTANCE	C-11
C.4.32	HS_FDS_FETCH_ROWS.....	C-11
C.4.33	HS_LANGUAGE.....	C-11
C.4.34	HS_NLS_NCHAR	C-11
C.4.35	LOG_DESTINATION	C-12
C.4.36	ORA_MAX_DATE	C-12
C.4.37	ORA_NLS33.....	C-12
C.4.38	ORACLE_DRDA_TCTL.....	C-12
C.4.39	ORACLE_DRDA_TRACE.....	C-13
C.4.40	TRACE_LEVEL	C-13

D National Language Support

D.1	Overview of NLS Interactions	D-2
D.2	Client and Oracle Integrating Server Configuration.....	D-4
D.3	Gateway Language Interaction with DRDA Server	D-4
D.3.1	Gateway Configuration.....	D-5
D.3.2	NLS Parameters in the Gateway Initialization File.....	D-5
D.4	Gateway Codepage Map Facility	D-7
D.5	Multi-Byte and Double-Byte Support in the Gateway.....	D-10
D.6	Message Availability	D-12
D.7	Example of NLS Configuration	D-12

E Configuration Worksheet

F Quick Reference to Oracle SQL Functions

G Sample Applications

G.1	DB2INS.....	G-2
G.2	ORAIND	G-4

Index

Send Us Your Comments

Oracle Transparent Gateway for DRDA Installation and User's Guide, 10g Release 1 (10.1) for UNIX

Part No. B12009-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us at the following e-mail address:

`infoibm_us@oracle.com`

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

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

Preface

The Oracle Transparent Gateway for DRDA provides users with transparent access to DRDA databases as if they were Oracle databases.

Intended Audience

This guide is intended for anyone responsible for installing, configuring, and administering the gateway, and also for application developers.

Read this guide if you are responsible for tasks such as:

- Installing and configuring the Oracle Transparent Gateway for DRDA
- Configuring the SNA and TCP/IP Products
- Setting up gateway security
- Diagnosing gateway errors
- Using the gateway to access tables in DRDA databases
- Writing applications that access DRDA databases through the gateway

You must understand the fundamentals of transparent gateways and the UNIX operating systems before using this guide to install or administer the gateway.

Processors

Refer to the *Oracle Database Installation Guide 10g for UNIX Systems* and to the certification matrix on Oracle MetaLink for the most up-to-date list of certified hardware platforms and operating system version requirements to operate the gateway for your Linux, AIX-Based, HP-UX, or Solaris system. The Oracle MetaLink web site can be found at the following URL:

<http://metalink.oracle.com/>

The gateway processor requirements for your platform are as follows:

- for Linux 32-bit: Intel Pentium-based processors
- for Linux 64-bit: Intel Itanium 2 based 64-bit systems
- for Linux S/390: Any processor that can run Linux S/390
- for AIX-Based Systems: IBM pSeries
- for HP-UX: HP 9000 Series HP-UX that can run the required version of HP-UX
- for Solaris: A Solaris Operating System (SPARC 64-bit) that can run the required version of Solaris with 64-bit architecture

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Related Documents

The *Oracle Transparent Gateway for DRDA Installation and User's Guide* (for UNIX) is included as part of your product shipment. Also included is:

Oracle Database Heterogeneous Connectivity Administrator's Guide

This guide contains information common to all heterogeneous gateways, including important information on functions, parameters, and error messages.

Oracle Database Administrator's Guide

Oracle Database Concepts

Oracle Database Error Messages

Oracle Database Performance Tuning Guide

Oracle Database Security Guide

Conventions

Examples of input and output for the gateway and Oracle environment are shown in a special font:

```
$ mkdir /ORACLE/your_name
```

All output is shown as it actually appears. For input, refer to the following list:

example text Words or phrases, such as `mkdir` and `ORACLE`, must be entered exactly as spelled and in the letter case shown. In this example, `mkdir` must be entered in lowercase letters and `ORACLE` must be entered in uppercase letters.

italic text Italicized uppercase and lowercase, such as *your_name*, indicates that you must substitute a word or phrase, such as an actual directory name.

BOLD text, or ***bold italic TEXT*** Bold words or phrases refer to a file directory structure, such as a directory, path, or file ID.

{ } Curly braces indicate that one of the enclosed arguments is required. Do not enter the curly braces themselves.

[] Brackets enclose optional clauses or arguments from which you can choose one or more. Do not enter the brackets themselves.

. . . Ellipses indicate that the preceding item can be repeated. You can enter an arbitrary number of similar items.

| The vertical line separates choices.

SQL*Plus Prompts

The SQL*Plus prompt, `SQL>`, appears in SQL statements and SQL*Plus command examples. Enter your response at the prompt. Do not enter the text of the prompt, "SQL>", in your response.

Storage Measurements

Storage measurements use the following abbreviations:

- KB, for kilobyte, which equals 1,024 bytes
- MB, for megabyte, which equals 1,048,576 bytes
- GB, for gigabyte, which equals 1,073,741,824 bytes

Introduction

The Oracle Transparent Gateway for DRDA enables you to:

- Integrate heterogeneous database management systems so that they appear as a single homogeneous database system
- Read and write data from Oracle applications to data in DB2/OS390, DB2/400, DB2 Universal Database, and DB2/VM databases in addition to any Oracle Database server data.
- Read this chapter for information about the architecture, uses, and features of the Oracle Transparent Gateway for DRDA.

This chapter contains the following sections:

- [Introduction to the Oracle Transparent Gateway](#) on page 1-2
- [Release 10g Gateways](#) on page 1-2
- [Gateway Capabilities](#) on page 1-3
- [Terms](#) on page 1-7
- [Architecture](#) on page 1-8
- [Implementation](#) on page 1-9
- [How the Gateway Works](#) on page 1-10
- [Oracle Tools and the Gateway](#) on page 1-10
- [Features](#) on page 1-11

1.1 Introduction to the Oracle Transparent Gateway

In today's global economy, information is a company's most valuable resource. Whether you need to analyze new markets, tailor your products to meet local demands, increase your ability to handle complex customer information, or streamline operations, your company requires instant access to current and complete information.

Company growth and diversification often mean functioning with a collage of applications and geographically scattered data that may be using incompatible networks, platforms, and storage formats. Diverse application standards and storage formats can make integration of information difficult. Oracle Corporation offers integration technologies to overcome these technical barriers. Oracle Open Gateways simplify complex systems and remove obstacles to information, providing your company the opportunity to focus on business.

1.1.1 Protection of Current Investment

Oracle Transparent Gateway for DRDA gives your company the ability to develop its information systems without forfeiting its investments in current data and applications. The gateway gives you access to your Oracle data and DB2 data with a single set of applications while you continue to use existing IBM applications to access your IBM data. You can also use more productive database tools and move to a distributed database technology without giving up access to your current data.

If you choose to migrate to Oracle Database technology and productivity, the gateway allows you to control the pace of your migration. As you transfer applications from your previous technology to the Oracle Database, you can use the gateway to move the DB2 data into Oracle Databases.

1.2 Release 10g Gateways

The Oracle Database 10g server provides the foundation for the next generation of the Oracle Open Gateways Release 10g, which will deliver enhanced integration capabilities by exploiting Oracle Database 10g Heterogeneous Services. Heterogeneous Services is a component of the Oracle Database 10g server. The Oracle Database 10g server provides the common architecture for future generations of the gateways. For detailed information on Oracle Heterogeneous Services, refer to *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

The version 10 gateways are even more tightly integrated with the Oracle Database 10g server than previous versions, enabling improved performance and enhanced functionality while still providing transparent integration of Oracle and non-Oracle data. For example, connection initialization information is available in the local Oracle Database 10g server, reducing the number of round trips and the amount of data sent over the network. SQL execution is also faster, because statements issued by an application are parsed and translated once and can then be reused by multiple applications.

Version 10 gateways leverage any enhancements in the Oracle Database 10g server, and you can quickly extend those benefits to your non-Oracle data.

1.2.1 Advantages of the Gateway

Oracle Transparent Gateway for DRDA enables Oracle applications to access the DRDA Application Servers, such as DB2 for MVS, through Structured Query Language (SQL). The gateway and the Oracle Database 10g server together create the appearance that all data resides on a local Oracle Database 10g server, even though data might be widely distributed. If data is moved from a DRDA Application Server database to an Oracle Database server, then no changes in application design or function are needed. The gateway handles all differences in both data types and SQL functions between the application and the database.

1.3 Gateway Capabilities

Oracle Transparent Gateway for DRDA gives you the power to integrate your heterogeneous system into a single, seamless environment. This integration enables you to make full use of existing hardware and applications throughout your corporate-wide environment. You can eliminate the need to rewrite applications for each configuration, and you can avoid the tedious, error-prone process of manual data transfer. Together with the Oracle tools, networking, and data server technology, the Oracle Transparent Gateway for DRDA sets a high standard for seamless, enterprise-wide information access.

Oracle Transparent Gateway for DRDA enables applications to read and update DB2 data and Oracle data as if all of the data were stored in a single database. As a result, end users and application programmers are not required to know either the physical location or the storage characteristics of the data. This transparency not only allows you to integrate heterogeneous data seamlessly, it simplifies your gateway implementation, application development, and maintenance.

1.3.1 Transparency at All Levels

The Oracle Transparent Gateway for DRDA gives you transparency at every level within your enterprise.

- Location transparency
End users can access tables by name without needing to understand the physical location of the tables.
- Network transparency
The gateways exploit the Oracle Net technology to allow users to access data across multiple networks without concern for the network architecture. TCP/IP protocol is supported.
- Operating system transparency
You can access data stored under multiple operating systems without being aware of the operating systems that hold the data.
- Data storage transparency
Data can be accessed regardless of the database or file format.
- Access method transparency
You can utilize a single dialect of SQL for any data store, eliminating the need to code for database-specific access methods or SQL implementations.

1.3.2 Extended Database Services

Following are some of the more sophisticated Oracle Database 10g server services available through the gateway.

- **SQL functions**
Your application can access all your data using Oracle SQL, which is rich in features. Advanced Oracle Database 10g server functions, such as outer joins, are available even if the target data stores do not support them in a native environment. The method by which the gateways are integrated with the Oracle Database 10g server ensures that the newest features of each database release are always available immediately to the gateway.
- **Distributed capabilities**
Heterogeneous data can be integrated seamlessly because Oracle distributed capabilities, such as JOIN and UNION, can be applied against non-Oracle data without any special programming or mapping.
- **Distributed query optimization**
The Oracle Database 10g server can utilize its advanced query optimization techniques to ensure that SQL statements are executed efficiently against any of your data. The data distribution and storage characteristics of local and remote data are equally considered.
- **Two-phase commit protection**
The Oracle server two-phase commit mechanism provides consistency across data stores by ensuring that a transaction that spans data stores is still treated as a single unit of work. Changes are not committed (or permanently stored) in any data store unless the changes can be committed in all data stores that will be affected.
- **Stored procedures and database triggers**
The same Oracle stored procedures and database triggers can be used to access all of your data, thereby ensuring uniform enforcement of your business rules across the enterprise.

1.3.3 Extended Advanced Networking, Internet and Intranet Support

The gateway integration with the Oracle Database 10g server extends (to non-Oracle data) the benefits of the Oracle Internet and Oracle Net software and extends the benefits of the Oracle client/server and server/server connectivity software. These powerful features include:

- **Application server support**
Any Internet or intranet application that can access data in Oracle can also incorporate information from data stores accessible through the gateways. Web browsers can connect to the Oracle Database using any application server product that supports Oracle software.
- **Implicit protocol conversion**
Oracle and Oracle Net can work together as a protocol converter, allowing applications to transparently access other data stores on platforms that do not support the client's network protocol. An Oracle Database 10g server can use TCP/IP to communicate with the gateway and another data store.

- **Advanced Security**

Non-Oracle data can be protected from unauthorized access or tampering during transmission to the client. This is done by using the hardware-independent and protocol-independent encryption and CHECKSUM services of Advanced Security.

1.3.4 Dynamic Dictionary Mapping

The simple setup of the gateway does not require any additional mapping. Before an application can access any information, the application must be told the structure of the data, such as the columns of a table and their lengths. Many products require administrators to manually define that information in a separate data dictionary stored in a hub. Applications then access the information using the hub dictionary instead of the native dictionaries of each database. This approach requires a great deal of manual configuration and maintenance on your part. As administrators, you must update the data dictionary in the hub whenever the structure of a remote table is changed.

Inefficient duplication is not necessary with Oracle Transparent Gateway for DRDA. The gateway uses the existing native dictionaries of each database. Your applications access data using the dictionaries designed specifically for each database, which means no redundant dictionary ever needs to be created or maintained.

1.3.5 SQL

Oracle Transparent Gateways ease your application development and maintenance by allowing you to access any data using a uniform set of SQL. Changes to the location, storage characteristics, or table structure do not require any changes to your applications. ANSI and ISO standard SQL are supported, along with powerful Oracle extensions.

1.3.6 Data Definition Language

Oracle Applications can create tables in target data stores by using native data definition language (DDL) statements.

1.3.7 Data Control Language

You can issue native data control language (DCL) statements from an Oracle environment, allowing central administration of user privileges and access levels for heterogeneous data stores.

1.3.8 Passthrough and Native DB2 SQL

Execution of native DB2 SQL can be passed through the gateway for execution directly against DB2. This enables applications to send statements, such as a DB2 CREATE TABLE, to the gateway for execution on a target DB2 system.

1.3.9 Stored Procedures

The gateway enables you to exploit both Oracle and non-Oracle stored procedures, leveraging your investments in a distributed, multi-database environment. Oracle stored procedures can access multiple data stores easily, without any special coding for the heterogeneous data access.

1.3.9.1 Oracle Stored Procedures

Oracle stored procedures enable you to access and update DB2 data using centralized business rules stored in the Oracle Database 10g server. Using Oracle stored procedures can increase your database performance by minimizing network traffic. Instead of sending individual SQL statements across the network, an application can send a single EXECUTE command to begin an entire PL/SQL routine.

1.3.9.2 Native DB2 Stored Procedures

The gateway can execute DB2 stored procedures using standard Oracle PL/SQL. The Oracle application executes the DB2 stored procedure as if it were an Oracle remote procedure.

1.3.10 Languages

Any application or tool that supports the Oracle Database 10g server can access over thirty different data sources through the Oracle gateways. A wide variety of open system tools from Oracle Corporation and third-party vendors can be used, even if the data is stored in legacy, proprietary formats. Hundreds of tools are supported, including ad hoc query tools, web browsers, turnkey applications, and application development tools.

1.3.11 Oracle Database server Technology and Tools

The gateway is integrated into the Oracle Database server technology, which provides global query optimization, transaction coordination for multi-site transactions, support for all Oracle Net configurations, and so on. Tools and applications that support the Oracle Database server can be used to access heterogeneous data through the gateway.

1.3.12 SQL*Plus

You can use SQL*Plus for moving data between databases. This product gives you the ability to copy data from your department databases to corporate Oracle Databases.

1.3.13 Two-Phase Commit and Multi-site Transactions

The gateway can participate as a partner in multi-site transactions and two-phase commit. How this occurs depends on the capabilities of the underlying data source, meaning that the gateway can be implemented as any one of the following:

- A full two-phase commit partner
- A commit point site
- A single-site update partner
- A read-only partner

The deciding factors for the implementation of the gateway are the locking and transaction-handling capabilities of your target database.

Oracle Transparent Gateway for DRDA, by default, is configured as a commit point site (that is, commit confirm protocol). Optionally, you can configure the gateway as read-only if you choose to enforce read-only capability through the gateway. Other protocols are not supported. Refer to ["Read-Only Gateway"](#) on page 13-5 in [Chapter 13, "Using the Gateway"](#).

1.3.14 Site Autonomy

All Oracle Database server products, including gateways, supply site autonomy. For example, administration of a data source remains the responsibility of the original system administrator. Site autonomy also functions such that gateway products do not override the security measures established by the data source or operating environment.

1.3.15 Migration and Coexistence

The integration of a data source through the gateway does not require any changes to be made to applications at the data source. The result is that the Oracle Database server technology is non-intrusive, providing coexistence and an easy migration path.

1.3.16 Security

The gateway does not bypass existing security mechanisms. Gateway security coexists with the security mechanisms already used in the operating environment of the data source.

Functionally, gateway security is identical to that of an Oracle Database server, as described in the *Oracle Database Administrator's Guide*. Oracle Database security is mapped to the data dictionary of the data source.

1.4 Terms

The terms used in this guide do not necessarily conform to IBM terminology. The following list presents several terms and their meanings as used within this guide:

DRDA data is, generically, any database data accessed through DRDA.

DRDA database is the collection of data that belongs to a DRDA Server

DRDA Server is a database server that can be accessed through DRDA. IBM terminology for a DRDA Server is a DRDA Application Server, or AS.

DRDA Server type is a specific database product or program that can act as a DRDA Server.

Oracle integrating server is any Oracle Database 10g server instance that communicates with the Oracle Transparent Gateway for DRDA to distribute database access operations to a DRDA Server. The Oracle integrating server can also be used for non-gateway applications.

DB2 Universal Database is a generic name for the UNIX-based or Windows-based implementations of DB2. DB2/UDB is frequently used as an abbreviation for DB2 Universal Database.

1.5 Architecture

The Oracle Transparent Gateway for DRDA works with the Oracle Database 10g server to shield most of the differences of the non-Oracle Database from Oracle applications. This means that the Oracle applications can access the Oracle Database 10g server data and the DRDA database data as if it were Oracle data located at the Oracle integrating server.

The architecture consists of the following main components:

- Client

The client is an Oracle application or tool.

- Oracle integrating server

The Oracle integrating server is an Oracle instance accessed by an Oracle Database 10g server with procedural and distributed options. Usually, the Oracle integrating server is installed on the same host as the gateway, but this is not a requirement. The Oracle integrating server and the gateway communicate in the normal Oracle Database server-to-server manner.

If the Oracle integrating server is not on the host where the gateway resides, then you must install the correct Oracle networking software on the platform where the server resides. For Oracle Database 10g, you must install Oracle Net on the Oracle Database 10g server machine.

- Oracle Transparent Gateway for DRDA

The gateway must be installed on hosts that are running the appropriate operating system.

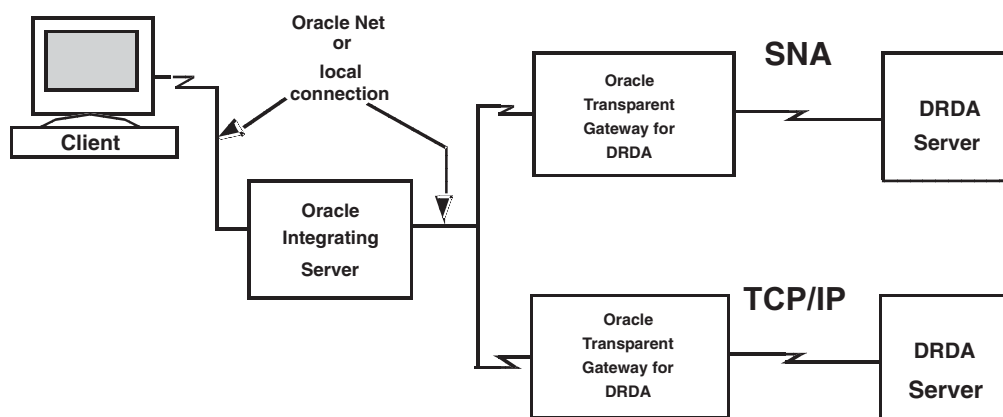
If the Oracle integrating server is not on the same host, then you must also install Oracle Net so that the gateway and Oracle Database 10g server can communicate.

- DRDA Server

The DRDA Server must be a DRDA Server database on a system accessible to the host and using either the SNA or TCP/IP protocols.

Multiple Oracle Database 10g servers can access the same gateway. A single host gateway installation can be configured to access more than one DRDA Server.

[Figure 1–1](#) illustrates the gateway architecture that is described above.

Figure 1–1 The Gateway Architecture

1.6 Implementation

When the gateway is installed on your host, it has some of the same components as an Oracle Database instance on your host. The gateway has the following components:

- A base file directory, similar to the one associated with an Oracle instance's ORACLE_HOME environment variable
- A gateway system identifier (SID), comparable to an Oracle instance's ORACLE_SID
- Oracle Net to support communication between the Oracle integrating server and the Oracle Transparent Gateway for DRDA

The gateway does not have:

- Control, redo log, or database files
- The full set of subdirectories and ancillary files that are associated with an installed Oracle Database 10g server

Because the gateway does not have background processes and does not need a management utility such as Oracle Enterprise Manager, you do not need to start the gateway product. Each Oracle Database 10g server user session that accesses a particular gateway creates an independent process on the host. This process runs the gateway session and executes SNA or TCP/IP functions to communicate with a DRDA Server.

1.7 How the Gateway Works

The gateway has no database functions of its own. Instead, it provides an interface by which an Oracle Database 10g server can direct part or all of a SQL operation to a DRDA database.

The gateway that is supporting the DRDA Server is identified to the Oracle integrating server using a database link. The database link is the same construct that is used to identify other Oracle Database 10g server databases. Tables on the DRDA Server are referenced in SQL as:

table_name@dblink_name

or

owner.table_name@dblink_name

If you create synonyms or views in the Oracle integrating server database, then you can refer to tables on the DRDA Server by using simple names as though the table were local to the Oracle integrating server.

When the Oracle integrating server encounters a reference to a table that is on the DRDA Server, the applicable portion of the SQL statement is sent to the gateway for processing. Any host variables that are associated with the SQL statement are bound to the gateway and therefore to the DRDA Server.

The gateway is responsible for sending these SQL statements to the DRDA Server for execution and for fielding and returning responses. The responses are either data or messages. Any conversions between Oracle datatypes and DRDA datatypes are performed by the gateway. Both the Oracle integrating server and the application read and process only Oracle datatypes.

1.7.1 SQL Differences

Not all SQL implementations are the same. The Oracle Database 10g server supports a larger set of built-in functions than the databases that are currently accessed through the gateway. The Oracle integrating server and the gateway work together to convert SQL to a form that is compatible with the specific DRDA Server.

During this conversion, an Oracle Database 10g server function can be converted to a function that is recognizable to the specific DRDA Server. For example, the Oracle Database 10g server NVL function is converted to the IBM VALUE function.

Alternatively, the Oracle integrating server withholds functions that are not executable by the DRDA Server and performs them after rows are fetched from the DRDA database. This processing generally applies to SELECT statements. The Oracle integrating server and the gateway cannot perform this kind of manipulation on UPDATE, INSERT, or DELETE statements because doing so changes transaction semantics.

1.8 Oracle Tools and the Gateway

Use the gateway to run applications, such as Oracle tools, that read and write data that is stored in DRDA databases.

While the Oracle Transparent Gateway for DRDA provides no new application or development facilities, it extends the reach of existing Oracle tools to include data in non-Oracle Databases that support DRDA.

Using the Oracle Transparent Gateway for DRDA with other Oracle products can greatly extend the capabilities of the stand-alone gateway. The following examples demonstrate how powerful the gateway is with other Oracle tools.

1.8.1 SQL*Plus

Use SQL*Plus and the Oracle Transparent Gateway for DRDA to create a distributed database system, providing an easy-to-use transfer facility for moving data between the distributed databases. One possible use is to distribute the data in your corporate Oracle Database to departmental DRDA databases. You can also distribute data in your corporate DRDA database to departmental Oracle Databases.

1.9 Features

Following is a list of important features that characterize this release of the gateway.

1.9.1 Heterogeneous Services Architecture

This release of the Oracle Transparent Gateway for DRDA utilizes the Oracle Heterogeneous Services component within the Oracle Database 10g server. Heterogeneous Services is the building block for the next generation of Oracle Open Gateways.

For detailed information about heterogeneous services, refer to *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

1.9.2 Performance Enhancements

Oracle Transparent Gateway for DRDA contains several internal performance enhancements. This product has shown major improvements in response time and CPU utilization for all relevant address spaces for a variety of workloads compared to version 9 gateways. The actual performance improvement at your site might vary, depending on your installation type and workload.

1.9.3 Fetch Reblocking

The array size of the application for SELECT is effective between the application and the Oracle integrating server. However, the array blocksize and the block fetch between the Oracle integrating server and the gateway are controlled by two Heterogeneous Services initialization parameters: HS_RPC_FETCH_SIZE and HS_RPC_FETCH_REBLOCKING. These parameters are specified in the Gateway Initialization File. Refer to *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information.

1.9.4 Oracle Database 10g Passthrough Supported

You can use the Oracle Database 10g DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE feature to pass commands or statements available in your DRDA database through the gateway.

1.9.5 Retrieving Result Sets Through Passthrough

Oracle Transparent Gateway for DRDA provides a facility to retrieve result sets from a select SQL statement issued with passthrough. Refer to ["Retrieving Results Sets Through Passthrough"](#) on page 14-29 for additional information.

1.9.6 Support for TCP/IP

This release of the gateway supports the TCP/IP communication protocol between the gateway and the DRDA Server. Refer to [Chapter 10, "Configuring TCP/IP"](#), for further information.

1.9.7 Native Semantics

This release of the gateway supports the ability to selectively enable or disable post-processing of various SQL functions by the DRDA Server. Refer to ["Native Semantics"](#) on page 14-19 for further information.

1.9.8 Columns Supported in a Result Set

Oracle Transparent Gateway for DRDA supports up to 1000 columns in a result set.

1.9.9 EXPLAIN_PLAN Improvement

The EXPLAIN_PLAN table contains the actual SQL statements passed to the DRDA Server from the Oracle Database 10g server through the gateway.

1.9.10 Heterogeneous Database Integration

The gateway support for ANSI-standard SQL enables read/write access to DRDA databases. Even if your data exists on different platforms in different applications, new applications can use all data, regardless of location.

1.9.11 Minimum Impact on Existing Systems

The gateway does not require installation of additional Oracle software on your OS/390 (MVS), AS/400, VM, or UNIX target system. The database interface that it uses is provided by IBM and is built into the DRDA database products and SNA or TCP/IP facilities that already exist on these platforms.

Configuring an IBM system for DRDA access typically consists of defining the SNA or TCP/IP resources involved and establishing access security definitions specific to the target database.

1.9.12 Large Base of Data Access

DRDA Application Server Function is supported by most IBM DB2 database products.

1.9.13 Application Portability

The gateway's ability to interface with heterogeneous databases makes it possible to develop a single set of portable applications that can be used against both Oracle and IBM databases, and any other databases for which Oracle Corporation provides gateways.

1.9.14 Remote Data Access

Location flexibility is maximized because the gateway architecture permits network connections between each of the components. The application can use the Oracle client-server capability to connect to a remote Oracle integrating server through Oracle Net. The Oracle integrating server can connect to a remote gateway using a database link. The gateway connects to DRDA Servers through SNA or TCP/IP network facilities.

The benefits of remote access are that it:

- Provides a means to allocate the appropriate resource to a given task
You can, for example, move application development off expensive processors and onto cost-efficient workstations or microcomputers.
- Expands the number of available data sources
Without remote access, you are limited to the data available in the local environment. With remote access, your data sources are limited only by your networks.
- Provides a means to tailor an application environment to a given user
For example, some users prefer a block-mode terminal environment, while others prefer a bit-mapped, graphics driven terminal environment. Remote access can satisfy both because you are not constrained by the interface environment imposed by the location of your data.

1.9.15 Support for Distributed Applications

Because the gateway gives your application direct access to DRDA data, you eliminate the need to upload and download large quantities of database data to other processors. Instead, you can access data where it is, when you want it, without having to move the data between machines and risk unsynchronized and inconsistent data. Avoiding massive data replication can also reduce aggregate disk storage requirements over all your systems.

However, if your system design requires moving data among the machines in a network, SQL*Plus and the gateway can simplify the data transfer. With a single SQL*Plus command, you can move entire sets of data from one node of the network to another and from one database to another.

You can pass commands and statements specific to your DRDA database through the gateway to be executed by the DRDA database. For example, you can pass DB2/OS390 commands through the gateway for DB2 to execute. You can also execute stored procedures defined in non-Oracle Databases.

1.9.16 Application Development and End User Tools

Through the gateway, Oracle Corporation extends the range of application development and end-user tools you can use to access your IBM databases. These tools increase application development and user productivity by reducing prototype, development, and maintenance time. Current Oracle users do not have to learn a new set of tools to access data stored in DRDA databases. Instead, they can access Oracle and DRDA data with a single set of tools.

With the gateway and the application development tools available from Oracle Corporation, you can develop a single set of applications to access Oracle and DRDA data. Users can use the decision support tools available from Oracle Corporation to access Oracle and DRDA data. These tools can run on remote machines connected through Oracle Net to the Oracle integrating server.

When designing applications, keep in mind that the gateway is designed for retrieval and relatively light transaction loads. The gateway is not currently designed to be a heavy transaction processing system.

1.9.17 Password Encryption Utility

This release of the gateway includes a utility to support encryption of plain-text passwords in the gateway Initialization File. Refer to [Chapter 15, "Security Considerations"](#) for details.

1.9.18 Support for DB2/OS390 V6 and V7 Stored Procedures

This release of the gateway supports the native stored procedure catalogs in DB2 V6 (SYSIBM.SYSROUTINES and SYSIBM.SYSPARMS).

1.9.19 Codepage Map Facility

This release of the gateway supports external mapping of IBM CCSIDs to Oracle character sets. Refer to ["Gateway Codepage Map Facility"](#) on page D-7 in [Appendix D, "National Language Support"](#).

1.9.20 IBM DB2 Universal Database Support

This release supports IBM DB2 Universal Database.

1.9.21 IBM DB2 Version 5.1 ASCII Tables

IBM DB2 Version 5.1 supports ASCII and EBCDIC character sets. The character set selection is defined during table creation. The Oracle Transparent Gateway for DRDA supports access to EBCDIC tables and ASCII tables. Refer to [Appendix D, "National Language Support"](#).

1.9.22 Read-Only Support

This release allows the gateway to be configured as a read-only gateway. In this mode, no modifying of user data will be allowed. For more information, refer to ["DRDA_READ_ONLY"](#) on page C-9.

Release Information

This chapter provides information specific to this release of the Oracle Transparent Gateway for DRDA. It includes the following sections:

- [Product Set](#) on page 2-2
- [Changes and Enhancements](#) on page 2-2
- [Bugs Fixed in Release 10](#) on page 2-2
- [Known Problems](#) on page 2-3
- [Known Restrictions](#) on page 2-4

2.1 Product Set

The following production components are included on the product CD-ROM:

Oracle Transparent Gateway for DRDA, release 10.1.0.2.0

Oracle Net, release 10.1.0.2.0

2.2 Changes and Enhancements

Following is a list of changes and enhancements unique to this release of the gateway.

Support for Graphic and Multi-byte Data

This release of the gateway adds support for DB2 GRAPHIC and VARGRAPHIC datatypes. Refer to [Chapter 14, "Developing Applications"](#).

Support for DB2/UDB on Intel Hardware

This release of the gateway adds support for DRDA Servers running on Microsoft Windows and Linux on Intel hardware.

Data Dictionary Support for DB2/UDB

This release of the gateway adds Oracle data dictionary support for DB2 UDB V7. Refer to ["Sample SQL scripts"](#) on page 12-7 in [Chapter 12, "Configuring the Gateway"](#).

2.3 Bugs Fixed in Release 10

3121041

ORA-28500, DRC=-30020, ERRP=GDJMRCCH, ERRMC=124C ON SELECT FROM AS/400

2854129

GATEWAY CRASHES WHEN INVALID DATE IS USED IN TO_DATE : ORA-2068, ORA-28511

2787316

INVALID DB2 PASSWORD RETURNS BACK ERRNC 124C INSTEAD OF 1245

1947548

QA - IN DOUBT TRANSACTION FOR SNA/DRDA WHEN DROPPING AND CREATING TABLES

1941672

QA - ORA-1821 CAN NOT INSERT USING TO_DATE () THROUGH GATEWAY

2.4 Known Problems

The problems that are documented in the following section are specific to the Oracle Transparent Gateway for DRDA, and are known to exist in this release of the product. These problems will be fixed in a future gateway release. If you have any questions or concerns about these problems, contact Oracle Support Services.

A current list of problems is available online. Contact your local Oracle Corporation office for information about accessing this online information.

Compatibility with DB2/UDB

This release of the gateway is not compatible with DB2/UDB V8. Contact Oracle Support Services and request status for bug number 3275652.

Bug No.: 3275652 During testing of Oracle Transparent Gateway for DRDA Release 9.2 and Release 10, while trying to bind, the following results occurred:

```
SQL> execute gtw$_bind_pkg@hgolink; BEGIN gtw$_bind_pkg@hgolink; END;

*
ERROR at line 1:
ORA-01403: no data found
TG4DRDA v10.1.0.2.0 grc=0, drc=-30020 (839C,0000), errp=GJDMRC, errmc=1245
ORA-06512: at line 1
```

SQL*Plus DESCRIBE command

The SQL*Plus DESCRIBE command cannot be used to access the remote DRDA Server object information.

2.5 Known Restrictions

The following restrictions are known to exist for the products in this release. Restrictions are not scheduled to change in future releases. Also refer to [Chapter 14, "Developing Applications"](#), for information or limitations when developing your applications.

Accessing DB2 Alias Objects

If you need to access DB2 alias objects on a remote DB2 system, then you must specify `DRDA_DESCRIBE_TABLE=FALSE` initialization parameter in the Gateway Initialization File.

Oracle SQL Command INSERT

When copying data from an Oracle Database server to a DRDA Server, the Oracle SQL command `INSERT` is not supported. The `SQL*Plus COPY` command must be used. Refer to [Chapter 13, "Using the Gateway"](#), for more information.

2.5.1 DB2 Considerations

DD Basic Tables and Views

The owner of DD basic tables and views is `OTGDB2`. This cannot be changed.

SUBSTR Function Post-Processed

The `SUBSTR` function can be used with the Oracle Database server in ways that are not compatible with a DRDA Server database, such as DB2/OS390. Therefore, the `SUBSTR` function is post-processed. However, it is possible to allow the server to process it natively using the ["Native Semantics"](#) feature. Refer to [Chapter 14, "Developing Applications"](#), for details.

AVS Mapping User IDs (DB2/VM)

APPC VTAM Support (AVS) has problems mapping user IDs that are sent using lowercase letters or special characters. Contact your IBM representative for additional information about this problem.

Support for DRDA Server Character Sets

Support for character sets used by a DRDA Server is configurable via the gateway's Codepage Map Facility. Refer to [Appendix D, "National Language Support"](#), for more information.

Datatype Limitations

Refer to ["DRDA Datatype to Oracle Datatype Conversion"](#) on page 14-22 for detailed information about datatypes.

SAVEPOINT Command Is Not Supported

Oracle Transparent Gateway for DRDA does not support the Oracle command `SAVEPOINT`.

Null Values and Stored Procedures

Null values are not passed into, or returned from, calls to stored procedures through the gateway.

String Concatenation of Numbers

String concatenation of numbers is not allowed in DB2/400, DB2/UDB, and DB2/OS390. For example, `2 || 2` is not allowed.

GLOBAL_NAMES Initialization Parameter

If GLOBAL_NAMES is set to TRUE in the Oracle Database server INIT.ORA file, then in order to be able to connect to the gateway, you must specify the Heterogeneous Services (HS) initialization parameter, HS_DB_DOMAIN, in the Gateway Initialization Parameter file to match the value of the DB_DOMAIN parameter of the Oracle Database server. Refer to [Chapter 12, "Configuring the Gateway"](#), for more information.

Binding the DRDA Package on DB2/UDB

The DRDA gateway package must be bound on the DRDA Server before the gateway can perform any SQL operations. Because of a DB2/UDB restriction, the ORACLE2PC table must be created in the DB2/UDB database before the package can be bound. For details, refer to [Chapter 12, "Configuring the Gateway"](#).

Date Arithmetic

In general, the following types of SQL expression forms do not work correctly with the gateway because of DRDA Server limitations:

```
date + number
number + date
date - number
date1 - date2
```

DRDA Servers do not allow number addition or subtraction with date datatypes. The date and number addition and subtraction (*date + number*, *number + date*, *date - number*) forms are sent through to the DRDA Server where they are rejected.

Also, DRDA Servers do not perform date subtraction consistently. When you subtract two dates (*date1 - date2*), differing interpretations of date subtraction in the DRDA Servers cause the results to vary by server.

Note: Avoid date arithmetic expressions in all gateway SQL until date arithmetic problems are resolved.

Row Length Limitation

Because of a restriction of the DRDA architecture, rows with aggregate length exceeding 32K bytes in DRDA representation cannot be stored or retrieved.

LONG Datatype in SQL*Plus

SQL*Plus cannot fetch LONG columns from the Oracle Transparent Gateway for DRDA.

Dictionary Views Are Not Provided for DB2/VM

Currently the Oracle Transparent Gateway for DRDA provides SQL for defining DB2/OS390, DB2/400, and DB2/UDB views that emulate parts of the Oracle Database dictionary. These are required for certain applications and tools that query dictionary tables. View definitions for DB2/VM are not provided in this release.

Single Gateway Instance per DRDA Network Interface

When installing the Gateway, a proper DRDA Network Interface must be chosen. Only one DRDA Network Interface may be chosen and installed per gateway instance. If the gateway product is re-installed, and if a Network Interface different from the previous installation is chosen, then the new choice will overlay the current installation.

Reconfiguration of the gateway Initialization Parameters must occur at this point in order to ensure proper gateway operation. If you wish to have both SNA and TCP/IP DRDA Network Interfaces installed, then you must create two separate gateway homes and installations.

2.5.2 SQL Limitations**Oracle ROWID Column**

DB2 does not have a column equivalent to the Oracle ROWID column. Because the ROWID column is not supported, the following restrictions apply:

- UPDATE and DELETE are not supported with the WHERE CURRENT OF CURSOR clause. To update or delete a specific row through the gateway, a condition style WHERE clause must be used. (Bug No. 205538)

When UPDATE and DELETE statements are used, in precompiler and PL/SQL programs, they rely internally on the Oracle ROWID function.

- Snapshots between Oracle Database servers and DB2 are not supported. Snapshots rely internally on the Oracle ROWID column.

Oracle Bind Variables

Oracle bind variables become SQL parameter markers when used with the gateway. Therefore, the bind variables are subject to the same restrictions as SQL parameter markers.

For example, the following statements are not allowed:

```
WHERE :x IS NULL
WHERE :x = :y
```

CONNECT BY Is Not Supported

Oracle Transparent Gateway for DRDA does not support CONNECT BY in SELECT statements.

System Requirements

This chapter provides information about hardware and software requirements that is specific to this release of the Oracle Transparent Gateway for DRDA. It includes the following sections:

- [Hardware Requirements](#) on page 3-2
- [Software Requirements](#) on page 3-3
- [Documentation Requirements](#) on page 3-4

3.1 Hardware Requirements

Includes processor, memory, network attachment, and drives

3.1.1 Processors

Refer to the *Oracle Database Installation Guide 10g for UNIX Systems* and to the certification matrix on Oracle MetaLink for the most up-to-date list of certified hardware platforms and operating system version requirements to operate the gateway for your Linux, AIX-Based, HP-UX, or Solaris system. The Oracle MetaLink web site can be found at the following URL:

<http://metalink.oracle.com/>

The gateway processor requirements for your platform are as follows:

- for Linux 32-bit: Intel Pentium-based processors
- for Linux 64-bit: Intel Itanium 2 based 64-bit systems
- for Linux S/390: Any processor that can run Linux S/390
- for AIX-Based Systems: IBM pSeries
- for HP-UX: HP 9000 Series HP-UX that can run the required version of HP-UX
- for Solaris: A Solaris Operating System (SPARC 64-bit) that can run the required version of Solaris with 64-bit architecture

3.1.2 Memory

For most installations, a minimum of 256 MB of real memory is recommended for the first user to support the Oracle Transparent Gateway for DRDA.

The total real memory requirement for each concurrent use of the gateway depends on the following factors:

- Number of concurrent APPC connections open by each user
- Number of concurrent TCP/IP connections open by each user
- Number of data items being transferred between the gateway and the remote transaction program
- Additional factors such as configured network buffer size

3.1.3 Network Attachment

The hardware requires any network attachment that is supported by either SNA server networking for SNA communication, or TCP/IP Networking Facility for TCP/IP communication. The network attachment for SNA is typically a Token Ring or SDLC Coaxial attachment. If you want concurrent SNA access, then the hardware must support independent LUs. The network attachment for TCP/IP is typically an Ethernet attachment.

3.1.4 CD-ROM Drive

An internal or external CD-ROM drive is required for installation.

3.1.5 Disk Space

Disk space required for installation:

- Solaris (64-bit) requires 1.25 GB
- AIX 5L requires 1.4 GB
- HP-UX 11.0 (64-bit) requires 1.2 GB
- Linux (Intel 32-bit and 64-bit) requires 400 MB
- Linux for S/390 requires 1.2 GB

3.2 Software Requirements

The system software configuration that is described in these requirements is supported by Oracle Corporation as long as the underlying system software products are supported by their respective vendors. Verify the latest support status with your system software vendors.

3.2.1 Operating System

- Linux for Intel Pentium-based 32-bit systems
- Linux for Intel Itanium 2 based 64-bit systems
- Linux for S/390
- Solaris 5.8 or later
- AIX 5L
- HP-UX 11.0 (64-bit)

3.2.2 DRDA Databases

You must have at least one of the following DRDA databases at a supported release level:

- DB2/OS390
- DB2/VM
- DB2/400
- DB2/Universal Database

3.2.3 Communication Server

The operating systems utilize specific communications servers.

3.2.3.1 Solaris

Sunlink SNA Peer-to-Peer Communications Server, Version 9

SNAP-IX, Version 6

3.2.3.2 AIX

IBM Communications Server for AIX, Version 6.0

3.2.3.3 HP

HPUX SNAPPlus2 Link Release 11.x and HPUX SNAPPlus2 API Release 11.x or higher are required.

3.2.4 Oracle Database server

The Oracle Database server which is to act as the Oracle integrating server requires the latest released patch set for Oracle Database 10g server release 10.1.0 or Oracle9i server release 9.2.

3.2.5 Oracle Networking Products

If the Oracle integrating server is not on the same host as the gateway, then Oracle Net is required to support communication between the host and the Oracle integrating server.

The following Oracle networking products are required on the same machine as the Oracle Database 10g Server:

- Oracle Net Client 10.1.0.2.0
- an Oracle Adapter version 10.1.0.2.0

Oracle Net software is included in this Oracle Transparent Gateway for DRDA release. Your gateway license includes a license for Oracle Net and an adapter of your choice. This license restricts the use of Oracle Net for gateway access.

3.3 Documentation Requirements

In addition to the documentation provided with the Oracle Transparent Gateway for DRDA distribution kit, the following Oracle documentation is recommended:

- *Oracle Database Administrator's Guide*
- *Oracle Database Application Developer's Guide - Fundamentals*
- *Oracle Database Heterogeneous Connectivity Administrator's Guide*
- *Oracle C++ Call Interface Programmer's Guide*
- *Oracle Call Interface Programmer's Guide*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database SQL Reference*
- *Oracle Net Services Administrator's Guide*
- *Oracle Net Services Reference Guide*

In addition to your Oracle documentation, ensure that you have appropriate documentation for your platform, for your operating system (OS), and for your DRDA Server (DB2/OS390, DB2/400, DB2 Universal Database, or DB2 server for VM).

The IBM publications regarding a distributed relational database might also be useful.

Installing the Gateway

This chapter provides general information about gateway installation that is specific to this release of the Oracle Transparent Gateway for DRDA. It contains the following sections:

- [Introduction](#) on page 4-2
- [Before You Begin](#) on page 4-2
- [Checklist for Gateway Installation](#) on page 4-3
- [Installation Overview](#) on page 4-4
- [Installing the Gateway from CD-ROM](#) on page 4-4
- [Installation Complete](#) on page 4-7

4.1 Introduction

The complete Oracle Transparent Gateway for DRDA installation process is divided into installation and configuration tasks. This process is described in Chapters 4 through 12. If this is the first time that the gateway has been installed on your host, then you must perform all of the steps that are documented in these chapters.

The installation tasks include:

- Ensuring that your hardware and software requirements are met
- Loading and installing the gateway software from the distribution medium into your system
- Determining your gateway system identifier
- Reconfiguring your network

An Installation Checklist follows, which you can use to check off each completed step in the process.

4.2 Before You Begin

This chapter requires you to input parameters that are unique to your system in order to properly configure the gateway. Refer to [Appendix E, "Configuration Worksheet"](#), for a worksheet listing all of the installation parameters that you will need to know in order to complete the configuration process. Ask your network administrator to provide these parameters before you begin.

You will also need to confirm that all hardware and software requirements have been met. Refer to [Chapter 3, "System Requirements"](#), to verify these requirements.

4.3 Checklist for Gateway Installation

Use the following checklist for installing the gateway:

- ☐ Step 1: Log on to the host
- ☐ Step 2: Create the product installation directory
- ☐ Step 3: Set the ORACLE_HOME environment variable
- ☐ Step 4: Mount the CD-ROM
- ☐ Step 6: Start the Oracle Universal Installer
- ☐ Step 7: Step through the Oracle Universal Installer
- ☐ Step 8: Verify installation success

4.4 Installation Overview

The primary installation tasks assume that you configure the gateway with a single Oracle integrating server and a single DRDA database. The steps for expanding the configuration to multiple integrating servers and multiple DRDA databases are described in [Chapter 12, "Configuring the Gateway"](#).

For general information about installing Oracle products, and how to use the Oracle Universal Installer, refer to the *Oracle Database Installation Guide 10g for UNIX Systems*.

4.5 Before Beginning Installation

Before installing the gateway, confirm that all hardware and software requirements are met. Refer to [Chapter 3, "System Requirements"](#), to verify these requirements.

4.6 Installing the Gateway from CD-ROM

The gateway is completely self-contained and must be installed in its own directory.

4.6.1 Step 1: Log on to the host

Log on to your host as the Oracle Database administrator (DBA) user. Refer to your platform release notes.

4.6.2 Step 2: Create the product installation directory

When you create a new directory, we recommend that you use the version number as part of the pathname. Doing so allows different versions of the same Oracle product to be installed under one Oracle directory tree. The product installation directory is also known as the ORACLE_HOME for the gateway, and it has also been called the tg4drda directory.

For example, enter:

```
$ mkdir /oracle
$ mkdir /oracle/tg4drda
$ mkdir /oracle/tg4drda/10.1.0
$ chown oracle:dba /oracle/tg4drda/10.1.0
$ chmod 755 /oracle/tg4drda/10.1.0
```

4.6.3 Step 3: Set the ORACLE_HOME environment variable

ORACLE_HOME must point to the directory that you created in Step 2. Set the ORACLE_HOME environment variable to point to this directory. The command that you enter depends on the shell that you are using.

For example, if you are a Bourne or Korn shell user, then enter:

```
$ ORACLE_HOME=/oracle/tg4drda/10.1.0; export ORACLE_HOME
```

If you are a C shell user, then enter:

```
$ setenv ORACLE_HOME /oracle/tg4drda/10.1.0
```

4.6.4 Step 4: Mount the CD-ROM

Place the CD-ROM in your CD-ROM drive:

4.6.4.1 Step 4a: on Solaris

Most Solaris installations will have the automounter running, in which case the CD-ROM will be mounted automatically. Typically the mount point will be `/cdrom`. To manually mount the CD-ROM, enter:

```
$ su root
# mkdir /cdrom
# mount -r -F hsfs /dev/dsk/c0t6d0s0 /cdrom
# exit
$ cd /cdrom
```

4.6.4.2 Step 4b: on AIX, enter:

```
$ su root
# mkdir /cdrom
# mount -rv cdrfs /dev/cd0 /cdrom
# exit
$ cd /cdrom
```

4.6.4.3 Step 4c: on HP-UX, enter:

```
$ su root
# nohup /usr/sbin/pfs_mountd &
# nohup /usr/sbin/pfsd &
# /usr/sbin/pfs_mount /SD_CDROM
# exit
$ cd /cdrom
```

4.6.4.4 Step 4d: on Linux, enter:

```
$ su root
# mkdir /cdrom
# mount /dev/hdb /cdrom
# exit
$ cd /cdrom
```

4.6.5 Step 5: Set the DISPLAY Variable

For example, if you are using a Bourne or Korn shell, then enter:

```
$ DISPLAY=machine:0; export DISPLAY
```

If you are a C shell user, then enter:

```
$ setenv DISPLAY machine:0
```

4.6.6 Step 6: Start the Oracle Universal Installer

The Oracle Universal Installer is provided on the distribution CD-ROM with the gateway. If you are installing over an older gateway instance, then you must upgrade the Oracle Universal Installer to this version by selecting the new Oracle Universal Installer from the Available Products menu.

For general information about installing Oracle products and how to use the Oracle Universal Installer, refer to the *Oracle Database Installation Guide 10g for UNIX Systems*.

Start the Installer with the following command:

```
$ ./runInstaller
```

4.6.7 Step 7: Step through the Oracle Universal Installer

Oracle Universal Installer is a menu-driven utility that guides you through installing the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform. Use the following list as a guide to the installation. The prompts that Oracle Universal Installer offers are listed in bold face font. Respond to the prompts by implementing the responses that follow the prompts.

Prompt: – File Locations: "Source and Destination"

Response: – Check that the Source Path points to the stage/products.jar file in the path of the mounted CD-ROM. Check that the Destination Path points to your ORACLE_HOME. Click "Next".

Prompt: – Available Products

Response: – Select "Oracle 10i Database". Click "Next".

Prompt: – Installation Types

Response: – Select "Custom". Click "Next".

Prompt: – Available Products Components

Response: – Open the "Oracle Transparent Gateways" product group and select "Oracle Transparent Gateway for DRDA". Remove selection from everything else for a standalone gateway installation. Click "Next".

Prompt: – Optional JDK home prompt

Response: – Click "Next".

Prompt: – DRDA Network Interface Product Software

Response: – Choose the network interface software appropriate for this installation of the gateway. Click "Next".

Prompt: – Summary

Response: – Verify the products to be installed. Click "Next".

4.6.8 Step 8: Verify installation success

After the Oracle Universal Installer confirms that the installation has ended, verify that the installation was successful. To do this, check the contents of the installation log file, located in the Oracle inventory's log directory. The default file name is **installActions<DATE>.log**.

Ignore the instruction to run the **root.sh** script, if applicable.

4.7 Installation Complete

Your gateway installation is now complete. Proceed with the configuration tasks described in Chapters 5 through 8.

4.7.1 De-installing the Gateway

De-installing the Oracle Transparent Gateway for DRDA requires the use of the Oracle Universal Installer. Follow the procedures below to de-install the gateway.

1. To restart the Oracle Universal Installer, refer to the installation process followed earlier in this chapter in ["Installing the Gateway from CD-ROM"](#) on page 4-4, and repeat the following steps (steps 1, 3, 4, and 6 of the installation startup process):
 - a. [Step 1: Log on to the host](#)
 - b. [Step 3: Set the ORACLE_HOME environment variable](#)
 - c. [Step 4: Mount the CD-ROM](#)
 - d. [Step 6: Start the Oracle Universal Installer](#)
2. When the **"Welcome"** panel appears, click the **"Deinstall Products"** button.
3. In the list of installed products, select the Gateway product and any other products you wish to remove, then click **"Remove."**

Configuring the DRDA Server

The steps for configuring your remote DRDA Server cover the following DRDA Servers:

- [Checklists for Configuring the DRDA Server](#) on page 5-2
- [DB2/OS390](#) on page 5-3
- [DB2/400](#) on page 5-4
- [DB2/UDB \(Universal Database\)](#) on page 5-5
- [DB2/VM](#) on page 5-7

Configuring a DRDA database to allow access by the gateway requires actions on the DRDA database and on certain components of the host operating system. Although no Oracle software is installed on the host system, access to (and some knowledge of) the host system and DRDA database are required during the configuration. Refer to the vendor documentation for complete information about your host system and DRDA database.

5.1 Checklists for Configuring the DRDA Server

Use the following checklists for configuring the DRDA Server.

5.1.1 DB2/OS390

- ☐ Step 1: Configure the Communications Server
- ☐ Step 2: Define the user ID that owns the package
- ☐ Step 3: Define the recovery user ID
- ☐ Step 4: Determine DRDA location name for DB2 instance
- ☐ Step 5: Configure DB2 Distributed Data Facility for gateway

5.1.2 DB2/400

- ☐ Step 1: Configure the Communications Server
- ☐ Step 2: Define the user ID that owns the package
- ☐ Step 3: Define the recovery user ID
- ☐ Step 4: Determine DRDA location name for DB2/400 instance

5.1.3 DB2/UDB (Universal Database)

- ☐ Step 1: Configure the SNA Communications Server
- ☐ Step 2: Define the user ID that owns the package
- ☐ Step 3: Define the recovery user ID
- ☐ Step 4: Determine DRDA location name for DB2/UDB instance

5.1.4 DB2/VM

- ☐ Step 1: Configure the Communications Server
- ☐ Step 2: Define the user ID that owns the package
- ☐ Step 3: Define the recovery user ID
- ☐ Step 4: Determine DRDA location name for DB2/VM instance

5.2 DB2/OS390

Experience with OS/390 (MVS), TSO, VTAM, and DB2 is required to perform the following steps:

5.2.1 Step 1: Configure the Communications Server

If you are using SNA, then configure OS/390 (MVS) VTAM for the SNA LU6.2 connection from the host. If you are using TCP/IP, then configure the TCP/IP subsystem, configure DB2's DDF subsystem to use TCP/IP, and assign a Primary and Recovery port number for the DB2 server.

5.2.2 Step 2: Define the user ID that owns the package

During gateway configuration, you will need to execute the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password that are used when the procedure is executed (either implied as the current Oracle user or explicitly defined in the CREATE DATABASE LINK command) must have proper authority on the DRDA Server to create the package. This same user ID should be used to create and own the ORACLE2PC (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have one or more of the following privileges on the DRDA Server:

- Package privileges of BIND, COPY, and EXECUTE, for example:

```
GRANT BIND    ON PACKAGE drda1.* TO userid
GRANT COPY    ON PACKAGE drda1.* TO userid
GRANT EXECUTE ON PACKAGE drda1.* TO PUBLIC
```

- Collection privilege of CREATE IN, for example:

```
GRANT CREATE IN ON PACKAGE drda1 TO USER userid
```

- System privileges of BINDADD and BINDAGENT, for example:

```
GRANT BINDADD TO USER userid
GRANT BINDAGENT TO USER userid
```

- Database privilege of CREATETAB, for example:

```
GRANT CREATETAB ON DATABASE database TO USER userid
```

Choose a user ID now that will own the package and the ORACLE2PC table. Ensure that this user ID is defined to both DB2 and OS/390 (MVS).

5.2.3 Step 3: Define the recovery user ID

During gateway configuration, the recovery user ID and password are specified in the Gateway Initialization File using the DRDA_RECOVERY_USERID and DRDA_RECOVERY_PASSWORD parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password defined in these parameters. This user ID must have execute privileges on the package and must be defined to the IBM DRDA database. If the user ID is not specified in DRDA_RECOVER_USERID, then the gateway attempts to connect to a user ID of ORARECOV when a distributed transaction is in doubt.

Determine the user ID and password you will use for recovery.

5.2.4 Step 4: Determine DRDA location name for DB2 instance

The DRDA location name is required as a gateway parameter. To determine the location name, issue the SQL query:

```
SELECT CURRENT SERVER FROM any_table
where any_table is a valid table with one or more rows.
```

If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact the system administrator to arrange to set a location name for the instance.

5.2.5 Step 5: Configure DB2 Distributed Data Facility for gateway

DB2 Distributed Data Facility (DDF) is the component of DB2 that manages all distributed database operations, both DRDA and non-DRDA.

If your site uses DB2 distributed operations, then DDF is probably operational on the DB2 instance you plan to access through the gateway. If DDF is not operational, then you must configure it and start it as described in the appropriate DB2 documentation.

Even if DDF is operational on the DB2 instance, it might be necessary to make changes to the DDF Communication Database (CDB) tables to specify the authorization conduct of DRDA sessions from the gateway. This can be done by properly authorized users with a utility like the DB2 SPUFI utility. If you make changes to CDB tables, then you must stop and restart DDF for the changes to take effect. Refer to [Chapter 15, "Security Considerations"](#), for additional CDB tables and security information.

5.3 DB2/400

Experience with DB2/400 and AS/400 is required to perform the following steps:

5.3.1 Step 1: Configure the Communications Server

If you are using SNA, then configure AS/400 communications for the SNA LU6.2 connection from the host. If you are using TCP/IP, then configure the TCP/IP subsystem, configure DB2/400 to use TCP/IP, and assign a Primary and Recovery port number for the DB2 server.

5.3.2 Step 2: Define the user ID that owns the package

During gateway configuration, you will need to execute the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password used when the procedure is executed (either implied as the current Oracle user or explicitly defined in the CREATE DATABASE LINK command) must have proper authority on the DRDA Server to create the package. This same user ID should be used to create and own the ORACLE2PC (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have the following privileges on the DRDA Server:

- Use authority on the CRTSQLPKG command
- Change authority on the library the package will be created in

Choose a user ID now that will own the package and ORACLE2PC table. Ensure that this user ID is defined to DB2/400 and AS/400.

5.3.3 Step 3: Define the recovery user ID

During gateway configuration, the recovery user ID and password are specified in the Gateway Initialization File using the DRDA_RECOVERY_USERID and DRDA_RECOVERY_PASSWORD parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password defined in these parameters. This user ID must have execute privileges on the package and must be defined to the IBM DRDA database. If the user ID is not specified in DRDA_RECOVER_USERID, then the gateway attempts to connect to a user ID of ORARECOV when a distributed transaction is in doubt.

Determine the user ID and password you will use for recovery.

5.3.4 Step 4: Determine DRDA location name for DB2/400 instance

The DRDA location name is required as a gateway parameter. To determine the location name, issue the following SQL query. If SQL is unavailable on the system, then use the AS/400 command DSPRDBDIRE to identify your "*LOCAL" DRDA Server.

```
SELECT CURRENT SERVER FROM any_table
where any_table is a valid table with one or more rows.
```

If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact the system administrator to arrange to set a location name for the instance.

5.4 DB2/UDB (Universal Database)

Experience with DB2/UDB, configuring the communication subsystem of DB2/UDB, and the host System Administration tools is required to perform the following steps.

5.4.1 Step 1: Configure the SNA Communications Server

If you are using SNA, then configure the communications server for the connection from the host. If you are using TCP/IP, then configure TCP/IP, configure DB2/UDB to use SNA and/or TCP/IP, and assign a Primary and Recovery port number for the DB2 server.

5.4.2 Step 2: Define the user ID that owns the package

During gateway configuration, you will need to execute the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password used when the procedure is executed (either implied as the current Oracle user or explicitly defined in the CREATE DATABASE LINK command) must have proper authority on the DRDA Server to create the package. This same user ID should be used to create and own the ORACLE2PC (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have one or more of the following privileges on the DRDA Server:

- Package privileges of BIND and EXECUTE, for example:

```
GRANT BIND      ON PACKAGE drda1.g2drsql TO USER userid
GRANT EXECUTE   ON PACKAGE drda1.g2drsql TO PUBLIC
```

- Schema privileges of CREATEIN, for example:

```
GRANT CREATEIN ON SCHEMA otgdb2 TO USER userid
GRANT CREATEIN ON SCHEMA drda1  TO USER userid
```

- Database authorities of CONNECT, BINDADD, and CREATETAB, for example:

```
GRANT CONNECT   ON DATABASE TO USER userid
GRANT BINDADD   ON DATABASE TO USER userid
GRANT CREATETAB ON DATABASE TO USER userid
```

Choose a user ID now that will own the package and ORACLE2PC table. Ensure that this user ID is defined to both the DB2 instance ID and AIX.

5.4.3 Step 3: Define the recovery user ID

During gateway configuration, the recovery user ID and password are specified in the Gateway Initialization File using the DRDA_RECOVERY_USERID and DRDA_RECOVERY_PASSWORD parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password defined in these parameters. This user ID must have execute privileges on the package and must be defined to the IBM DRDA database. If the user ID is not specified in DRDA_RECOVER_USERID, then the gateway attempts to connect to a user ID of ORARECOV when a distributed transaction is in doubt.

Determine the user ID and password you will use for recovery.

5.4.4 Step 4: Determine DRDA location name for DB2/UDB instance

The DRDA location name is required as a gateway parameter. To determine the location name, issue the SQL query:

```
SELECT CURRENT SERVER FROM any_table
where any_table is a valid table with one or more rows.
```

If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact your system administrator to arrange to set a location name for the instance.

5.5 DB2/VM

Experience with VM, AVS, and DB2/VM is required to perform the following steps:

5.5.1 Step 1: Configure the Communications Server

If you are using SNA, then configure VM VTAM and AVS for the SNA connection from the host. If you are using TCP/IP, then configure the TCP/IP Service.

5.5.2 Step 2: Define the user ID that owns the package

During gateway configuration, you will need to execute the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password used when the procedure is executed (either implied as the current Oracle user or explicitly defined in the CREATE DATABASE LINK command) must have proper authority on the DRDA Server to create the package. This same user ID should be used to create and own the ORACLE2PC (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have the following privileges on the DRDA Server:

- Package privileges of BIND, COPY, and EXECUTE
- Collection privilege of CREATE IN
- System privileges of BINDADD and BINDAGENT

Choose a user ID now that will own the package and ORACLE2PC table. Ensure that this user ID is defined to DB2/VM and VM.

5.5.3 Step 3: Define the recovery user ID

During gateway configuration, the recovery user ID and password are specified in the Gateway Initialization File using the DRDA_RECOVERY_USERID and DRDA_RECOVERY_PASSWORD parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password defined in these parameters. This user ID must have execute privileges on the package and must be defined to the IBM DRDA database. If the user ID is not specified in DRDA_RECOVER_USERID, then the gateway attempts to connect to a user ID of ORARECOV when a distributed transaction is in doubt.

Determine the user ID and password you will use for recovery.

5.5.4 Step 4: Determine DRDA location name for DB2/VM instance

The DRDA location name is required as a gateway parameter. To determine the location name, issue the SQL query:

```
SELECT CURRENT SERVER FROM any_table
where any_table is a valid table with one or more rows.
```

If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact the system administrator to arrange to set a location name for the instance.

Configuring SunLink for the Gateway

This chapter describes configuring the SunLink SNA Peer-to-Peer product on Solaris for use with the Oracle Transparent Gateway for DRDA. SunLink provides SNA connectivity via the APPC/LU6.2 protocol between the Sun host and the remote DRDA Server. Read this chapter to learn more about creating server profiles. host

This chapter contains the following sections:

- [Checklist for Configuring the Communications Interfaces](#) on page 6-2
- [Step 1: Setting up a Gateway Name](#) on page 6-3
- [Step 2: Setting up a Configuration File](#) on page 6-3
- [Starting the SunLink Peer-to-Peer Version 9 Software](#) on page 6-4
- [Step 3: Side Information File](#) on page 6-4
- [Step 4: Test the Connection](#) on page 6-5

6.1 Checklist for Configuring the Communications Interfaces

- ☐ [Step 1: Setting up a Gateway Name](#)
- ☐ [Step 2: Setting up a Configuration File](#)
- ☐ [Step 3: Side Information File](#)
- ☐ [Step 4: Test the Connection](#)

6.2 Step 1: Setting up a Gateway Name

The gateway name plays an important role in the use of the SunLink software. The gateway name is how users of SunLink client programs (for Oracle, it is the Oracle gateway kernel) identify the gateway.

After you decide on a gateway name on your Sun host, you can define it in one of two ways:

Method 1:

Use NIS/NIS+ to create the gateway name in the NIS/NIS+ database. Refer to the *SunLink Runtime and System Administrator's Guide* for an example.

Method 2:

Use a flat file, `/etc/appc`, to define the gateway name in your workstation.

For example, enter:

```
sunlinkgtw sunhost:sunlinkgtw
```

where:

`sunlinkgtw` is the gateway name.

`sunhost` is the hostname of the Sun workstation.

6.3 Step 2: Setting up a Configuration File

To enable communication between a SunLink gateway and a remote SNA host, you must specify (in SNA terms) the precise configuration of a SunLink gateway on your Sun host. The information is contained in a flat ASCII file, `./appc`.

For SunLink Version 9.0, this `./appc` file is input to the `sunpu2.1` utility.

The input file has the form of verbs with associated parameters. Each of the verb identifiers corresponds to an Application Programming Interface (API) verb. Depending on the hardware configuration of your SNA network, this input file might have different verbs and parameters.

Refer to the *SunLink Runtime and System Administrator's Guide* for a detailed description of each verb in the input file and its associated parameters.

A sample SNA configuration file for SunLink Version 9 is shipped with the gateway. After you have successfully installed the Oracle Transparent Gateway for IBM DRDA, you can find it in the `tg4drda/sna/sunlink` subdirectory. The sample file is:

- `$ORACLE_HOME/tg4drda/sna/sunlink/SunLink9.cfg`, for `sunpu2.1` configuration connecting the Sun host to several IBM hosts

Also shipped with the gateway are sample Side Information Files. They are also located in the `tg4drda/sna/sunlink` subdirectory as outlined in the following list:

- `$ORACLE_HOME/tg4drda/sna/sunlink/DB251ALU`, a Version 9 sample Side Information File
- `$ORACLE_HOME/tg4drda/sna/sunlink/DB2V23LU`, a Version 9 sample Side Information File

- `$ORACLE_HOME/tg4drda/sna/sunlink/DRDA400`, a Version 9 sample Side Information File
- `$ORACLE_HOME/tg4drda/sna/sunlink/DB2VM51`, a Version 9 sample Side Information File

6.3.1 Starting the SunLink Peer-to-Peer Version 9 Software

To start the SunLink Version 9 software, start the gateway by issuing the command:

```
sunpu2.1 -f ./appc
```

where `./appc` is the name of the configuration file.

6.4 Step 3: Side Information File

Before starting an APPC conversation with a partner program, a CPI-C program requires certain information. This information, known as side information, is provided by the Side Information File. The symbolic destination name corresponds to an entry in the Side Information File containing the `Partner_LU_name`, `Mode_name`, and `TP_name`.

6.4.1 Partner_LU_name

This identifies the name of the remote, or partner LU associated with the DRDA Server. In addition to specifying the fully qualified partner LU name, you can also specify a partner LU alias to identify a partner LU location profile (if required).

To allow more than one concurrent conversation between the gateway and the DRDA Server, specify that parallel sessions are supported in this profile.

6.4.2 Mode_name

This must be defined in the communication software of the DRDA Server. DRDA Servers use the mode name `IBMRDB` in many DRDA examples, but this is not required. Choose the mode name and the other mode parameters after consulting the person responsible for configuring the DRDA Server-side communications software. The `mode_name` you specify must exist at the target DRDA Server. It does not need to be defined in the same SNA gateway.

The parameters, related to parallel session limits, play a role in determining the maximum number of concurrent conversations allowed between a gateway instance and the DRDA Server. This equates to the maximum number of open database links using the gateway instance.

6.4.3 TP_name

This generally identifies the program to be executed on the server side of an APPC conversation. DRDA uses a special reserved TPN (called an SNA Service Transaction Program) that is expressed in hexadecimal because it contains non-printable characters. The TPN for DB2/MVS, DB2/UDB, and DB2/400 is `X'07F6C4C2'`.

For DB2/VM, the DRDA Server does not use the standard DRDA TPN. Instead, the TPN identifies the VM Resource ID (RESID) of the target DB2/VM server virtual machine, and can be non-hexadecimal characters. The RESID is specified when DB2/VM is configured.

6.4.4 Sample Side Information File for Version 9

The name of the Side Information File for Version 9 must be specified in the Gateway Initialization File **initsid.ora** as:

```
DRDA_CONNECT_PARM=/oracle/tg4drda/side9/DB2V23LU
```

Enter the value in the [Appendix E, "Configuration Worksheet"](#).

Here is a sample Side Information File for SunLink Version 9.0:

```
PTNR_LU_NAME = DB2V23
MODE_NAME    = IBMRDB
TP_NAME      = x'07F6C4C2'
```

6.5 Step 4: Test the Connection

Before proceeding with the installation and configuration of the Oracle Transparent Gateway for DRDA, test that your connection is working. You can do this using `sunop` or `sungmi`. Refer to your Sunlink documentation for more details.

6.6 Using SNA Session Security Validation

When the database link request for the gateway begins, the gateway attempts to start an APPC conversation with the DRDA Server. Before the conversation can begin, a session must start between the host Logical Unit (LU) and the DRDA Server LU.

SNA and its various access method implementations (including SunLink and VTAM) provide security validation at session initiation time, allowing each LU to authenticate its partner. This is carried out entirely by network software before the gateway and server application programs begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in the host Connection Profile and in similar parameter structures in the DRDA Server system that is to be accessed. Refer to the appropriate SunLink product documentation for detailed information.

6.7 SNA Conversation Security

SNA conversation security is determined by the setting of the gateway initialization parameter, `DRDA_SECURITY_TYPE`. This parameter determines whether SNA security option `SECURITY` is set to `PROGRAM` or to `SAME`. Generally, the gateway operates under SNA option `SECURITY=PROGRAM`, but it can also be set to operate under SNA option `SECURITY=SAME`.

6.7.1 SNA Security Option `SECURITY=PROGRAM`

If `DRDA_SECURITY_TYPE=PROGRAM` is specified, then the gateway allocates the conversation with SNA option `SECURITY=PROGRAM` and sends this information to the user ID:

- If the database link has explicit `CONNECT` information, then the specified user ID and password are sent.
- If the database link has no `CONNECT` clause and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID and password are sent.

- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit CONNECT information, then no user ID and password are sent. If no user ID and password are sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

In general, SECURITY=PROGRAM tells the DRDA Server to authenticate the user ID/password combination using whatever authentication mechanisms are available. For example, if DB2/OS390 is the DRDA Server, then RACF can be used. This is not always the case, however, because each of the DRDA Servers can be configured to process inbound user IDs in other ways.

6.7.2 SNA Security Option SECURITY=SAME

If DRDA_SECURITY_TYPE=SAME is specified, then the gateway allocates the conversation with SNA option SECURITY=SAME, and the following information is sent to the DRDA Server:

- If the database link has explicit CONNECT information, then the specified user ID is sent.
- If the database link has no CONNECT clause, and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID is sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit CONNECT information, then no user ID is sent. If no user ID is sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

For this option to function properly, SunLink requires that the effective user ID under which the gateway is executing must be a member of the system group. In UNIX terms, this means that the user ID must be defined with its primary group set to system. In addition, the owning user ID of the gateway executable must be set to the desired effective user ID, and the set-uid bit of the executable file permissions must also be set. The `ls -l` command shows the owning user ID and the setting of the set-uid bit for the executable file. The owning user ID can be changed by the root user with the `chown` command, and the set-uid bit can be set using the `chmod u+s` command. The gateway executable, as installed by the Oracle Universal Installer, has its set-uid bit disabled.

The simplest way to cause the gateway to execute under an effective user ID that is a member of the system group is to change the owning user ID of the gateway executable to `root`. Another way is to change the primary group for the owning user ID of the gateway executable to `system`. However, be careful when choosing the user ID. Oracle Corporation recommends using `root` and recommends never changing the Oracle dba user ID primary group to `system`.

When the effective user ID is not a member of the system group, a failure is generated when the gateway attempts to allocate a conversation with the DRDA Server, and an error message is sent to the gateway user.

Configuring SNAP-IX Interfaces

This chapter describes configuring the SNAP-IX product on Solaris for usage with the Oracle Transparent Gateway for IBM DRDA. SNAP-IX provides SNA connectivity via the APPC/LU6.2 protocol between the Sun host and the remote DRDA Server. Read this chapter to learn more about creating server profiles.

This chapter contains the following sections:

- [Steps for Configuring the Communications Interfaces](#) on page 2
- [SNAP-IX Configuration Tool](#) on page 2
- [Creating SNAP-IX Profiles for the Gateway](#) on page 2
- [Independent Versus Dependent LUs](#) on page 2
- [Creating SNA Definitions for the Gateway](#) on page 3
- [Sample SNAP-IX Definitions](#) on page 3
- [Configuring SNAP-IX](#) on page 3
- [Testing the Connection](#) on page 15

7.1 Steps for Configuring the Communications Interfaces

1. Create SNAP-IX profiles for the gateway
2. Create SNA definitions for the gateway
3. Test the connection

7.2 Before You Begin

This chapter requires you to input parameters unique to your system in order to properly configure SNAP-IX. Refer to [Appendix E](#) for a worksheet listing all of the installation parameters you will need to know before you can complete the configuration process. Ask your network administrator to provide you with these parameters before you begin.

7.3 SNAP-IX Configuration Tool

All SNAP-IX product configuration is done using the `xsnaadmin` program. This tool is an X-Windows application which provides a graphical interface so that you can view and modify the current SNAP-IX configuration and the current running state of the host SNA node.

7.4 Creating SNAP-IX Profiles for the Gateway

The Oracle Transparent Gateway for IBM DRDA requires a stored set of definitions, called Side Information Profiles, to support connections between the gateway and DRDA Servers. Each profile consists of a profile name and a profile type, which is a set of fields describing the profile. The fields in a given profile type are generally a mixture of operating parameter values and names of other SNA profiles relevant to the profile. Each functional part of APPC, such as the Mode, Remote Transaction Program name, and Logical Unit (LU), is described by a distinct profile type.

7.5 Independent Versus Dependent LUs

The Gateway configuration can accommodate either independent LUs or dependent LUs. If you choose to use dependent LUs, or are restricted to using dependent LUs, the Gateway will function properly; if a dependent LU is correctly defined, then you will need to make no alterations to the configuration of the Oracle Transparent Gateway for IBM DRDA, nor should any changes be needed to the DRDA Server. However, Oracle Corporation recommends using independent LUs for the Oracle Transparent Gateway for IBM DRDA because they support multiple parallel sessions or conversations. This means that multiple Oracle client applications can be active simultaneously with the same DRDA Server through the independent LU.

In contrast to independent LUs, dependent LUs support only a single active session. The CP (Control Point for the Node) queues each additional conversation request from the gateway behind an already active conversation. In other words, conversations are single-threaded for dependent LUs.

The operational impact of dependent LUs is that the first client application can initiate a conversation through the gateway with the DRDA Server, but while that session is active (which could be seconds, minutes or hours, depending on how the client application and transaction are designed), any other client application initiating a session with the same DRDA Server appears to hang as it waits behind the previous session.

If a production application really uses only a single conversation at any one time, then there should be no problem. However, at some point you might require additional concurrent conversations for testing or for other application development. Having more than one conversation requires that additional dependent LUs be defined on the remote host. Additional configuration entries will need to be added to SNAP-IX. Additional Side Information Profiles should be defined to use the new dependent LUs. Oracle Transparent Gateway for IBM DRDA instances should be created and configured to use these new Side Information Profiles.

7.6 Creating SNA Definitions for the Gateway

SNAP-IX definitions are stored in the following two files, located in the directory `/etc/opt/sna`:

- `sna_node.cfg` - SNA node definitions
- `sna_domn.cfg` - SNA domain definitions

These files are created and maintained with the `xsnaadmin` tool. Maintenance of SNA definitions is normally done by a user with administrative authority. The following information is intended for the person creating SNA definitions for the gateway. You should have some knowledge of SNA before reading this section.

7.6.1 Sample SNAP-IX Definitions

The `$ORACLE_HOME/tg4drda/sna/snapix` subdirectory contains a set of sample SNAP-IX definitions for the gateway, created with the `xsnaadmin`. SNA definitions are very specific to the host and SNA network. As such, the sample definitions provided will not work without being tailored for the local host and SNA network.

7.6.2 Configuring SNAP-IX

This section describes the process of creating your SNA definitions for SNAP-IX, using `xsnaadmin`. All of the tasks described in this section are performed from within `xsnaadmin`.

All configuration is done using the various pull-down menus and panels in `xsnaadmin`. The following configuration descriptions follow the samples provided. Please tailor the various SNA values for your local host and SNA network.

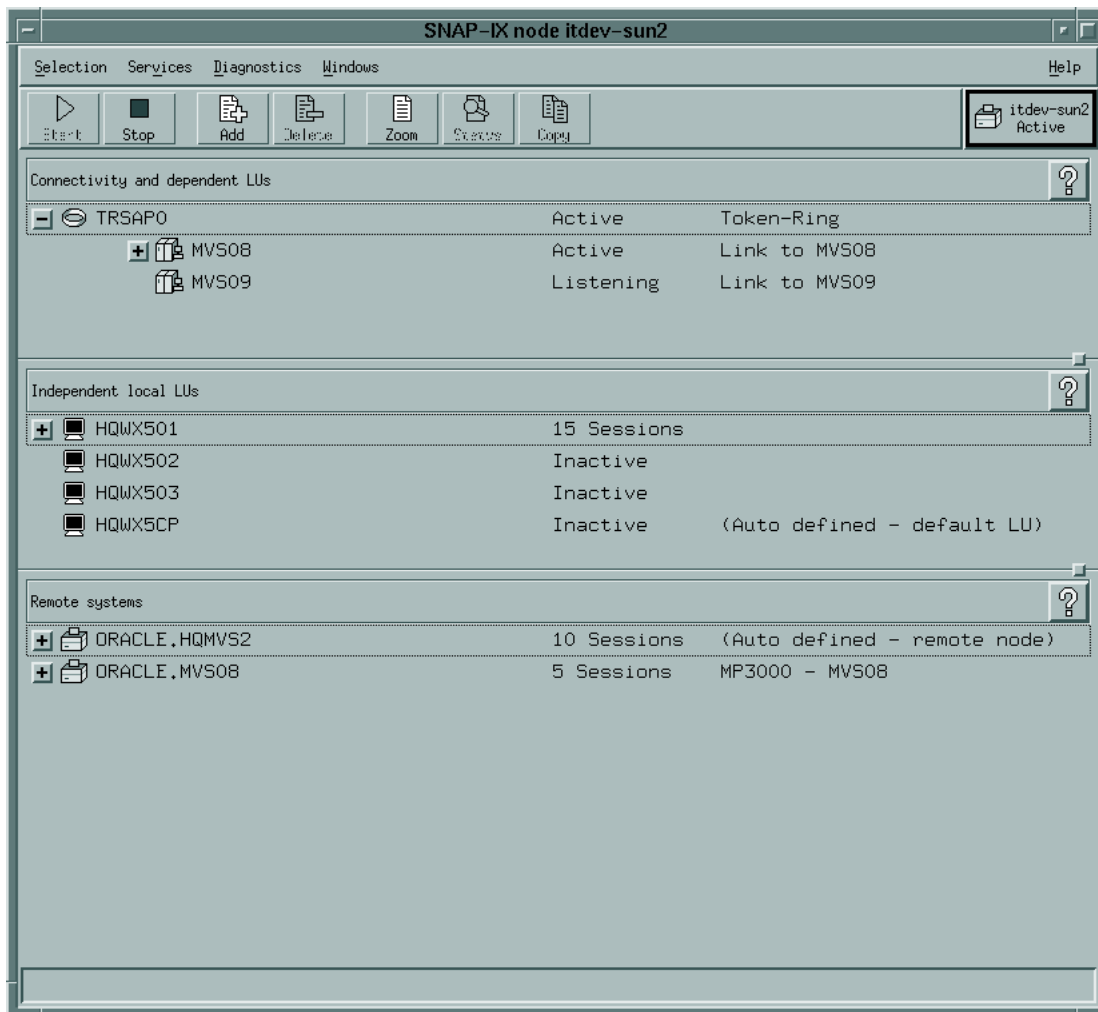
7.6.3 Invoking `xsnaadmin`

Use the following commands to invoke `xsnaadmin`. The `$DISPLAY` environmental variable must be set appropriately. If you are running `xsnaadmin` from the local console, then `$DISPLAY` should already be set. If you are running `xsnaadmin` from a remote X display, then set `$DISPLAY` to the host name or IP address of that display.

```
$ DISPLAY=xstation10.us.oracle.com:0
$ export DISPLAY
$ xsnaadmin &
```

Upon startup of `xsnaadmin`, the main screen will open and display the current configuration of the local SNA node. (See [Figure 7-1](#))

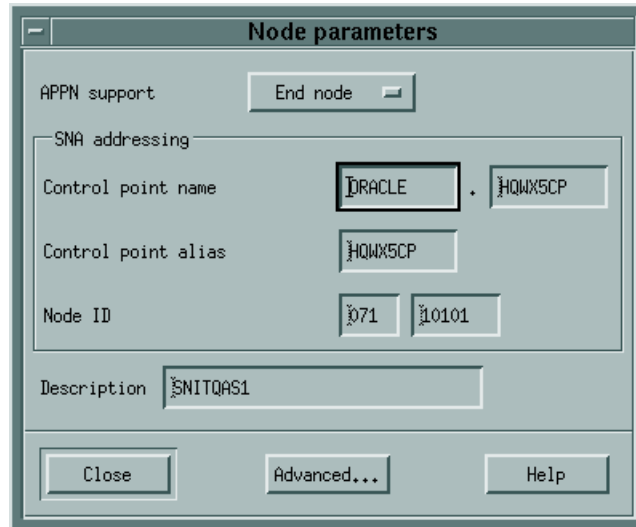
Figure 7-1 *xsnaadmin* Main Screen



Configuring the SNA node

From the Services menu select Configure Node Parameters. In the Node Parameters dialog box (see [Figure 7-2](#)) enter the APPN support type, Control Point Name, Control Point Alias and Node ID as needed. The Control Point Name is composed of the SNA Network Name and the CP name of the local host. Click **OK**.

Figure 7-2 Node Parameters Dialog Box



Adding a Port

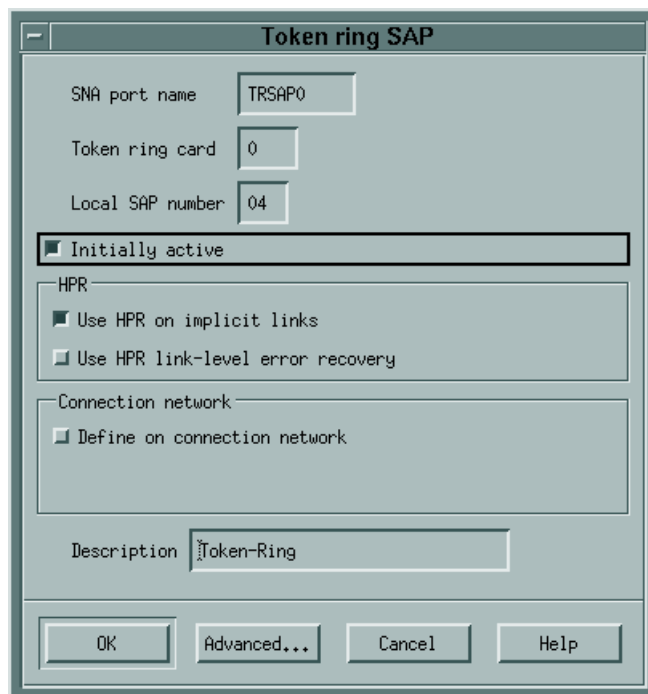
From the Services menu select Connectivity and New Port. In the Add to <nodename> dialog box ([Figure 7-3](#)), select the Port type and click **OK**.

Figure 7-3 Add to <nodename> Dialog Box



In the SAP dialog box (see [Figure 7-4](#)) enter a Port name and network card number. The Port name will be used to logically name the physical network card that you are using and will be used to bind a Service Access Port to the card for SNA protocols. Normally you can accept the values provided in the dialog box. If a different network card is needed, however, enter the card number as reported with the `dmesg` command. Click **OK**.

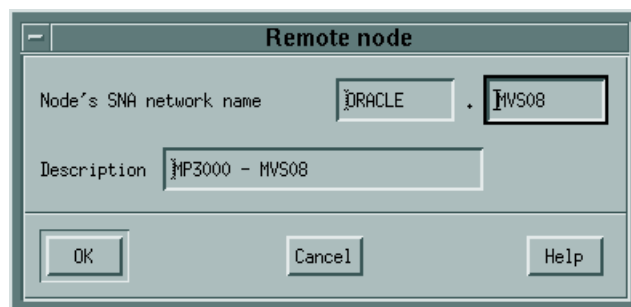
Figure 7-4 Token-ring SAP Dialog Box



Create a Link Station

Once the Port has been defined, you need to create a Link Station. The Link Station represents the SNA node of the remote host of the DRDA Server. But before you can create the Link Station, you must create a Remote Node definition. From the Services menu select APPC and Add Remote Node. In the dialog box (see [Figure 7-5](#)) enter the SNA CPNAME of the remote node and click **OK**.

Figure 7-5 Add Remote Node Dialog Box



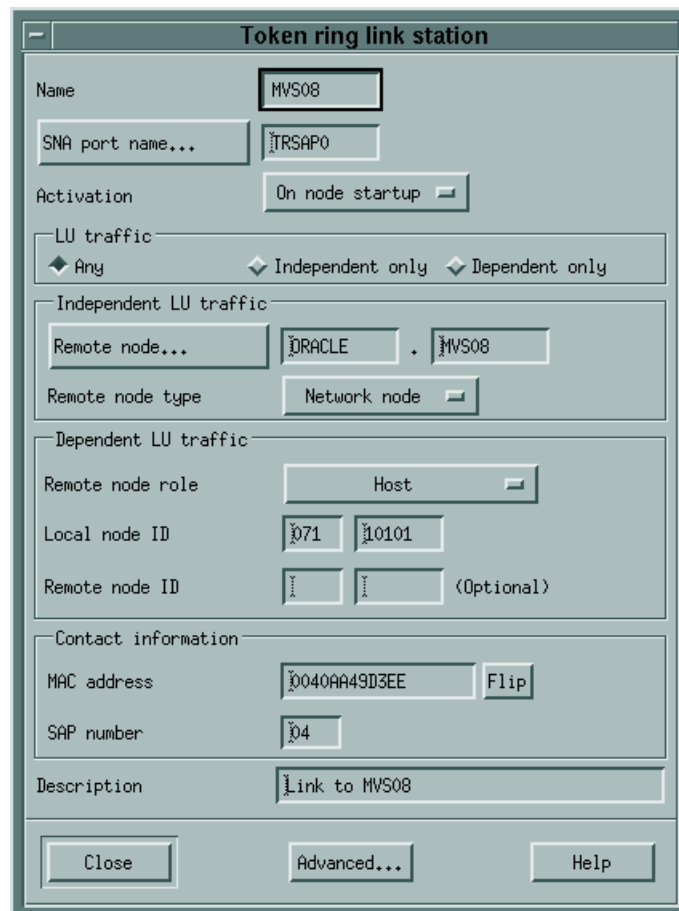
Now you are ready to create the Link Station. From the Services menu, select Connectivity and New Link Station. In the dialog box (see [Figure 7–6](#)) select the Port previously defined and click **OK**.

Figure 7–6 Add Link Station Dialog Box



In the Link Station dialog box (see [Figure 7–7](#)) enter a name for the Link Station, choose the SNA Port name and choose the type of link activation. Choose the LU Traffic type. For maximum flexibility, choose the Any option. For Independent LU traffic, specify the Remote Node name. Click on **Remote Node** and select the node you previously created. Click **OK**. Choose the type of the Remote node, typically a Network node. For Dependent LU traffic, specify the role of the Remote node, typically 'host', the Local Node ID, and optionally, Remote Node ID. Then specify the Contact Information.

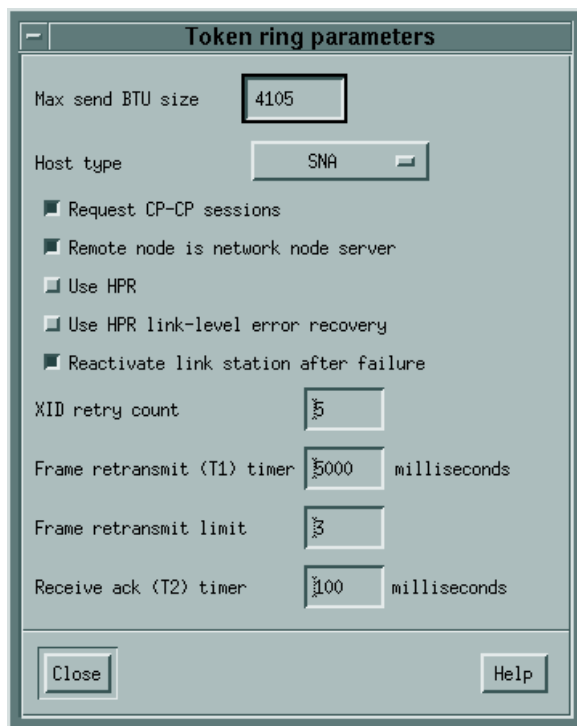
Figure 7–7 Link Station Dialog Box



Contact information contains the MAC address of the remote host as well as the SAP number. Press the **Advanced** button for additional parameters of the Link Station.

The Token Ring Parameters dialog box shows additional parameters of the Link Station (see [Figure 7-8](#)). These parameters effect initial XID contact and retransmission times and limits. The defaults are normally sufficient. Click **OK**.

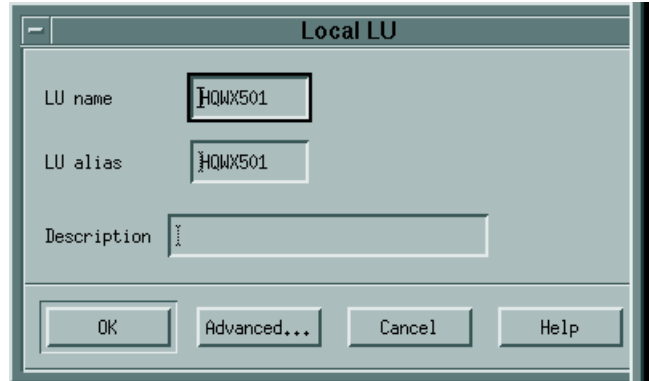
Figure 7-8 *Token Ring Parameters Dialog Box*



Create Local LUs

Once the Remote Node definitions have been made, create the Local LU names for the local host. From the Services menu select APPC and New Local LU. In the dialog box (see [Figure 7-9](#)) enter the name of the local LU and an alias. This name must correspond to the VTAM definitions on the remote DRDA Server host for the UNIX host. Click **OK**.

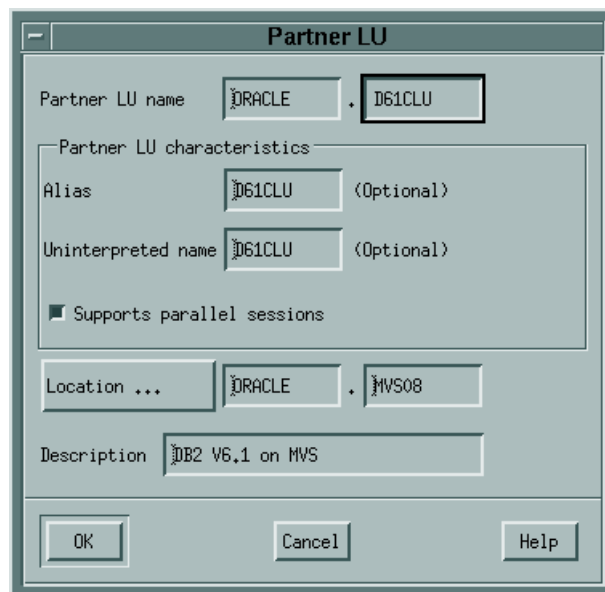
Figure 7-9 Local LU Dialog Box



Create Partner LUs

Now define a Partner LU which represents the LU that the DRDA Server is using to communicate. From the Services menu select APPC and New Partner LUs and Partner LU on Remote Node. In the dialog box (see [Figure 7-10](#)) Enter the Partner LU name and characteristics. The Partner LU name will contain the SNA Network Name as well as the LU name of the remote LU. Enable parallel session support. The location is the name as the Remote Node name. You may click on **Location** for a list. Then click **OK**.

Figure 7-10 Partner LU Dialog Box

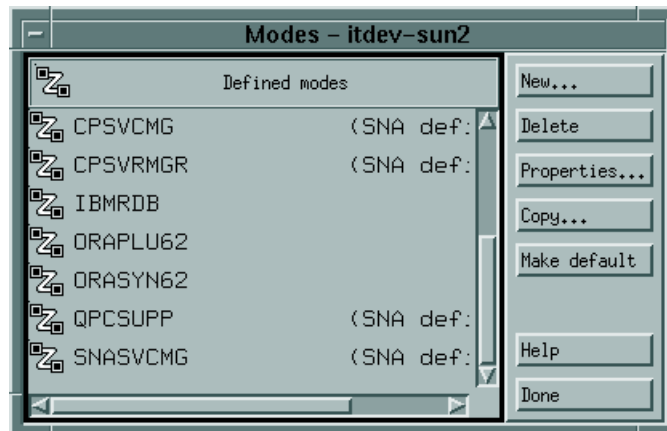


Create Mode and CPI-C Profiles

Once the local and remote LU definitions have been made, create the necessary Mode and CPI-C definitions.

From the Services menu select APPC and Modes. In the Modes dialog box (see [Figure 7-11](#)) click on **New** to add a new mode.

Figure 7-11 Modes Dialog Box



In the Mode dialog box (see [Figure 7-12](#)) enter the Mode Name and other session parameters. The prescribed name for a DRDA mode is "IBMRDB". Contact your Remote Host system administrator for appropriate mode parameters. Click **OK**.

Figure 7-12 IBMRDB Mode Dialog Box

Mode

Name:

Session limits:

Initial: Maximum:

Min con. winner sessions: Min con. loser sessions:

Auto-activated sessions:

Receive pacing window:

Initial: Maximum: (Optional)

☐ Specify timeout

☐ Restrict max RU size

Description:

OK Cancel Help

Now that the Mode has been defined, create the CPI-C Side Information Profile, which the gateway will use as a connection name. From the menu select APPC and CPI-C. In the CPI-C destination names dialog box (see [Figure 7-13](#)) click on **New** to add a new Profile.

Figure 7-13 CPI-C destination Names Dialog Box

CPI-C destination names

CPI-C symbolic destination names

- APPCMVS
- CICSPGA
- D51CLU DB2 V5.1 on MVS
- D51FLU DB2 V5.1 on MVS
- D61CLU DB2 V6.1 on MVS**
- D61FLU DB2 V6.1 on MVS
- D71CLU DB2 V7.1 on MVS

New... Delete Properties... Copy... Help Done

In the CPI-C destination dialog box (see [Figure 7-14](#)) enter the Profile name, Local LU name, Partner TP, Partner LU and mode, and Security option. The default TP name of the mode **DRDA Server** will typically be a Service TP named "07F6C4C2". For the Local LU, you may specify a specific LU or choose the default LU. For the Partner LU, enter either the full LU name or the alias created previously. Enter "IBMRDB" for the mode name. Choose the type of security these sessions will use. This will affect how session authorization is done. Click **OK**.

Figure 7-14 CPI-C destination Dialog Box

CPI-C destination

Name:

Local LU

◆ Specify local LU alias:

◆ Use default LU

Partner LU and mode

◆ Use PLU alias:

◆ Use PLU full name

Mode:

Partner TP

◆ Application TP

◆ Service TP (Hex):

Security

◆ None ◆ Same ◆ Program ◆ Program strong

User ID:

Password:

Description:

OK Cancel Help

7.7 Using SNA Session Security Validation

When the database link request for the gateway begins, the gateway attempts to start an APPC conversation with the DRDA Server. Before the conversation can begin, a session must start between the host Logical Unit (LU) and the DRDA Server LU.

SNA and its various access method implementations (including SNAP-IX and VTAM) provide security validation at session initiation time, allowing each LU to authenticate its partner. This is carried out entirely by network software before the gateway and server application programs begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in the host Connection Profile and in similar parameter structures in the DRDA Server system that is to be accessed. Refer to the appropriate SNA server product documentation for detailed information.

7.8 SNA Conversation Security

SNA conversation security is determined by the setting of the gateway initialization parameter, `DRDA_SECURITY_TYPE`. This parameter determines whether SNA security option `SECURITY` is set to `PROGRAM` or to `SAME`. Generally, the gateway operates under SNA option `SECURITY=PROGRAM`, but it can also be set to operate under SNA option `SECURITY=SAME`.

7.8.1 SNA Security Option `SECURITY=PROGRAM`

If `DRDA_SECURITY_TYPE=PROGRAM` is specified, then the gateway allocates the conversation with SNA option `SECURITY=PROGRAM` and sends this information to the user ID:

- If the database link has explicit `CONNECT` information, then the specified user ID and password are sent.
- If the database link has no `CONNECT` clause and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID and password are sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID and password are sent. If no user ID and password are sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

In general, `SECURITY=PROGRAM` tells the DRDA Server to authenticate the user ID/password combination using whatever authentication mechanisms are available. For example, if DB2/OS390 is the DRDA Server, then RACF can be used. This is not always the case, however, because each of the IBM DRDA Servers can be configured to process inbound user IDs in other ways.

7.8.2 SNA Security Option SECURITY=SAME

If `DRDA_SECURITY_TYPE=SAME` is specified, then the gateway allocates the conversation with SNA option `SECURITY=SAME`, and the following information is sent to the DRDA Server:

- If the database link has explicit `CONNECT` information, then the specified user ID is sent.
- If the database link has no `CONNECT` clause, and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID is sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID is sent. If no user ID is sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

For this option to function properly, SNAP-IX requires that the effective user ID under which the gateway is executing must be a member of the system group. In UNIX terms, this means that the user ID must be defined with its primary group set to `system`. In addition, the owning user ID of the gateway executable must be set to the desired effective user ID, and the set-uid bit of the executable file permissions must also be set. The `ls -l` command shows the owning user ID and the setting of the set-uid bit for the executable file. The owning user ID can be changed by the root user with the `chown` command, and the set-uid bit can be set using the `chmod u+s` command. The gateway executable, as installed by the Oracle Universal Installer, has its set-uid bit disabled.

The simplest way to cause the gateway to execute under an effective user ID that is a member of the system group is to change the owning user ID of the gateway executable to `root`. Another way is to change the primary group for the owning user ID of the gateway executable to `system`. However, be careful when choosing the user ID. Oracle Corporation recommends using `root` and recommends never changing the Oracle dba user ID primary group to `system`.

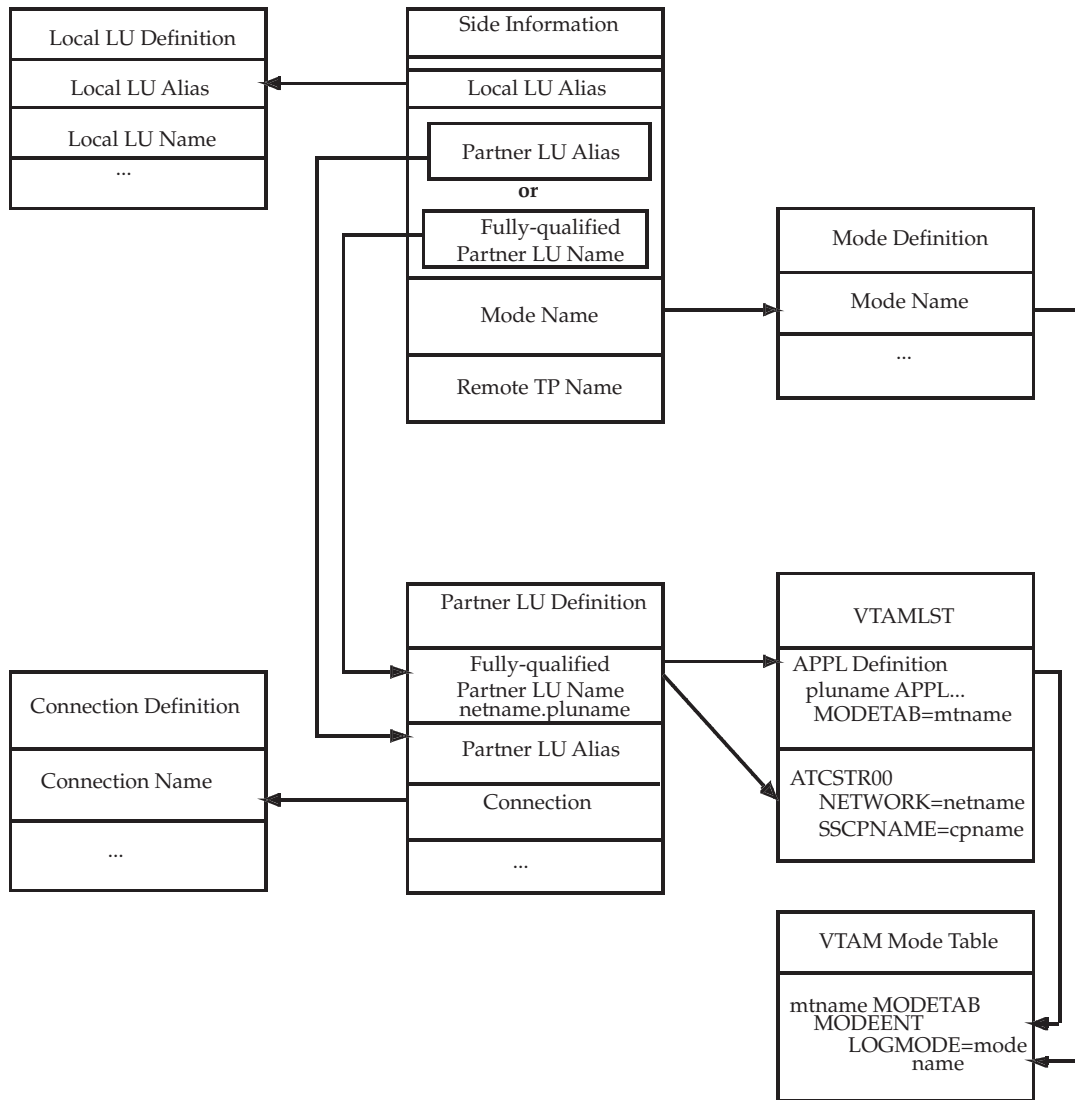
When the effective user ID is not a member of the system group, a failure is generated when the gateway attempts to allocate a conversation with the DRDA Server, and an error message is sent to the gateway user.

7.9 Testing the Connection

Before proceeding with the gateway configuration tasks in [Chapter 12, "Configuring the Gateway"](#), ensure that your connection is working. Do this by starting the SNAP-IX Node and then starting the individual link stations.

[Figure 7-15](#) shows the relationship between SNAP-IX definitions and the VTAM definitions on the remote host.

Figure 7-15 Relationship between SNAP-IX Definitions and Host VTAM Definitions



Configuring IBM Communication Server

This chapter describes configuring the IBM Communication Server product on AIX for usage with the Oracle Transparent Gateway for DRDA. IBM Communication Server provides SNA connectivity via the APPC/LU6.2 protocol between the host and the remote DRDA Server. Read this chapter to learn more about creating server profiles.

The following topics are included:

[Checklist for Configuring the Communications Interfaces](#) on page 8-2

[Step 1: Configuring Communication Server Profiles](#) on page 8-2

[Step 2: Creating Communication Server Profiles for the Gateway](#) on page 8-2

[Step 3: Testing the Connection](#) on page 8-5

[Using SNA Session Security Validation](#) on page 8-5

[SNA Conversation Security](#) on page 8-5

8.1 Checklist for Configuring the Communications Interfaces

- ❑ [Step 1: Configuring Communication Server Profiles](#)
- ❑ [Step 2: Creating Communication Server Profiles for the Gateway](#)
- ❑ [Step 3: Testing the Connection](#)

8.2 Step 1: Configuring Communication Server Profiles

Configure the profiles to define APPC conversations with DRDA databases.

8.3 Step 2: Creating Communication Server Profiles for the Gateway

Communications Server requires a stored set of definitions, called profiles, to support connections between the gateway and DRDA Servers. Each profile consists of a profile name and a profile type, a set of fields describing the profile. The fields in a given profile type are generally a mixture of operating parameter values and names of other SNA profiles relevant to the profile. Each functional part of APPC, such as the Mode, Remote Transaction Program name, and Logical Unit (LU), is described by a distinct profile type.

SNA profile definitions are created and modified in two ways:

- directly with shell commands
- using menus in the AIX System Management Interface Tool (SMIT)

Maintenance of SNA profiles is normally done by a user with root authority.

8.3.1 Sample Profile Definitions

The `$ORACLE_HOME/tg4drda/sna/commsvr` subdirectory contains a set of sample IBM Communication Server definitions for the gateway, created with the SMIT administration tool. SNA definitions are very specific to the host and SNA network. As such, the sample definitions provided will not work without being tailored for the local host and SNA network.

Before building the SNA profiles, examine these files to determine requirements. The export file format is text-oriented, and each field of each profile is labeled. You can print a copy of the export file to use while working with your profiles in a SMIT session.

8.3.2 Profile Types

There are different types of Communications Server profiles relevant to gateway APPC/LU6.2 operation. You can create and edit profiles by using a corresponding SMIT menu reached from the "Communications Applications and Services" primary menu.

The profiles relevant to the gateway are presented here in hierarchical order. Those profile types that are lowest in the hierarchy are discussed first. This matches the logical sequence for creating the profiles. You can use the SMIT "list" pop-up menu to fill in profile names.

8.3.2.1 Mode Profile

The Mode Profile specifies parameters that determine:

- APPC/LU6.2 parallel session limits
- send and receive pacing values
- SNA RU size
- the mode name that is sent to the server at session initiation

The mode name that you specify must be defined in the DRDA Server's communication software. DRDA Servers use the mode name IBMRDB in many DRDA examples, but this is not required. Choose the mode name and the other mode parameters after consulting the person responsible for configuring the DRDA Server-side communications software.

The parameters (related to parallel session limits) play a role in determining the maximum number of concurrent conversations allowed between a gateway instance and the DRDA Server. This equates to the maximum number of open database links using the gateway instance.

8.3.2.2 Local LU Profile

The Local LU Profile describes the SNA LU through which the gateway communicates. The LU type field must be LU6.2. The network name is an established name for your SNA network.

An LU name must be assigned to the gateway. The LU name assigned to the gateway might be required elsewhere in the SNA network. Contact the person responsible for your SNA network to determine the correct network and LU name to specify in the profile.

Set the dependent LU field to "no". Setting the dependent LU field to "yes" prevents more than one instance of the gateway from running at the same time.

The Local LU Profile name is specified in the Side Information Profile.

8.3.2.3 Link Profiles

The Link Profiles describe and control the connection of the host to the network. The gateway does not impose special requirements on these profiles.

The Link Profile name is specified in the Local LU Profile.

8.3.2.4 Partner LU Profile

The Partner LU Profile identifies the name of the remote, or partner, LU associated with the DRDA Server. In addition to specifying the fully qualified partner LU name, you can also specify a partner LU alias to identify a Partner LU Location Profile (if required).

To allow more than one concurrent conversation between the gateway and the DRDA Server, specify that parallel sessions are supported in this profile.

8.3.2.5 Partner LU Location Profile

The Partner LU Location Profile is only required when the target OLTP resides on a non-APPN (advanced peer-to-peer networking) node. The profile is also required if the owning node of the network connection to the target OLTP system is a non-APPN node.

In the mainframe environment, APPN support requires VTAM Version 4. Prior releases of VTAM are not APPN-enabled.

In configurations where the Partner LU Location Profile is required, the fully qualified partner owning CP name in the profile should be set to the value specified in the VTAM SSCPNAME start parameter. The Partner LU Location Profile name is specified as the alias in the Partner LU Profile.

8.3.2.6 Side Information Profile

The Side Information Profile is the top of the hierarchy for APPC/LU6.2 conversations. The name of the Side Information Profile is specified with DRDA_CONNECT_PARM in the gateway initialization file.

Enter the following information in the Side Information Profile fields:

Field – Local LU Name

Specification – LU Name as specified in local LU profile

Field – Fully Qualified Partner LU Name or partner LU alias

Specification – LU Name specified in Partner LU profile or Partner LU location profile

Field – Mode Name

Specification – Mode Name specified in Mode profile

Field – Remote Transaction Program Name

Specification – Remote TPN

Field – Remote TPN in hexadecimal

Specification – Yes

The Remote Transaction Program Name (TPN) generally identifies the program to be executed on the server side of an APPC conversation. IBM DRDA uses a special reserved TPN (called an SNA Service Transaction Program) that is expressed in hexadecimal because it contains non-printable characters. The TPN is X'07F6C4C2'. Specify this TPN for DB2/MVS and DB2/400 DRDA Servers.

For DB2/VM, the DRDA Server does not use the standard DRDA TPN. Instead, the TPN identifies the VM Resource ID (RESID) of the target DB2/VM server virtual machine and can be entered in non-hexadecimal characters. The RESID is specified when DB2/VM is configured.

8.4 Step 3: Testing the Connection

Before proceeding with the gateway configuration tasks in [Chapter 12, "Configuring the Gateway"](#), ensure that your connection is working. This can be done using SMIT.

8.5 Using SNA Session Security Validation

When the database link request for the gateway begins, the gateway attempts to start an APPC conversation with the DRDA Server. Before the conversation can begin, a session must start between the host Logical Unit (LU) and the DRDA Server LU.

SNA and its various access method implementations (including IBM Communication Server and VTAM) provide security validation at session initiation time, allowing each LU to authenticate its partner. This is carried out entirely by network software before the gateway and server application programs begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in the host Connection Profile and in similar parameter structures in the DRDA Server system that is to be accessed. Refer to the appropriate SNA server product documentation for detailed information.

8.6 SNA Conversation Security

SNA conversation security is determined by the setting of the gateway initialization parameter, `DRDA_SECURITY_TYPE`. This parameter determines whether SNA security option `SECURITY` is set to `PROGRAM` or to `SAME`. Generally, the gateway operates under SNA option `SECURITY=PROGRAM`, but it can also be set to operate under SNA option `SECURITY=SAME`.

8.6.1 SNA Security Option `SECURITY=PROGRAM`

If `DRDA_SECURITY_TYPE=PROGRAM` is specified, then the gateway allocates the conversation with SNA option `SECURITY=PROGRAM` and sends this information to the user ID:

- If the database link has explicit `CONNECT` information, then the specified user ID and password are sent.
- If the database link has no `CONNECT` clause and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID and password are sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID and password are sent. If no user ID and password are sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

In general, `SECURITY=PROGRAM` tells the DRDA Server to authenticate the user ID/password combination using whatever authentication mechanisms are available. For example, if DB2/OS390 is the DRDA Server, then RACF can be used. This is not always the case, however, because each of the DRDA Servers can be configured to process inbound user IDs in other ways.

8.6.2 SNA Security Option SECURITY=SAME

If `DRDA_SECURITY_TYPE=SAME` is specified, then the gateway allocates the conversation with SNA option `SECURITY=SAME`, and the following information is sent to the DRDA Server:

- If the database link has explicit `CONNECT` information, then the specified user ID is sent.
- If the database link has no `CONNECT` clause, and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID is sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID is sent. If no user ID is sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

For this option to function properly, IBM Communications Server requires that the effective user ID under which the gateway is executing must be a member of the system group. In UNIX terms, this means that the user ID must be defined with its primary group set to `system`. In addition, the owning user ID of the gateway executable must be set to the desired effective user ID, and the set-uid bit of the executable file permissions must also be set. The `ls -l` command shows the owning user ID and the setting of the set-uid bit for the executable file. The owning user ID can be changed by the root user with the `chown` command, and the set-uid bit can be set using the `chmod u+s` command. The gateway executable, as installed by the Oracle Universal Installer, has its set-uid bit disabled.

The simplest way to cause the gateway to execute under an effective user ID that is a member of the system group is to change the owning user ID of the gateway executable to `root`. Another way is to change the primary group for the owning user ID of the gateway executable to `system`. However, be careful when choosing the user ID. Oracle Corporation recommends using `root` and recommends never changing the Oracle dba user ID primary group to `system`.

When the effective user ID is not a member of the system group, a failure is generated when the gateway attempts to allocate a conversation with the DRDA Server, and an error message is sent to the gateway user.

Configuring SNAPPlus2

This chapter describes configuring the SNAPPlus2 product on HP-UX for usage with the Oracle Transparent Gateway for DRDA. SNAPPlus2 provides SNA connectivity via the APPC/LU6.2 protocol between the HP9000 host and the remote DRDA Server. Read this chapter to learn more about creating server profiles.

This chapter contains the following sections:

- [Steps for Configuring the Communications Interfaces](#) on page 9-2
- [Before You Begin](#) on page 9-2
- [SNAPPlus2 Configuration Tool](#) on page 9-2
- [Creating SNAPPlus2 Profiles for the Gateway](#) on page 9-2
- [Independent Versus Dependent LUs](#) on page 9-2
- [Creating SNA Definitions for the Gateway](#) on page 9-3
- [Using SNA Session Security Validation](#) on page 9-13
- [SNA Conversation Security](#) on page 9-13
- [Testing the Connection](#) on page 9-14

9.1 Steps for Configuring the Communications Interfaces

1. Create SNAPPlus2 profiles for the gateway
2. Create SNA definitions for the gateway
3. Test the connection

9.2 Before You Begin

This chapter requires you to input parameters unique to your system in order to properly configure SNAPPlus2. Refer to [Appendix E](#) for a worksheet listing all of the installation parameters you will need to know before you can complete the configuration process. Ask your network administrator to provide you with these parameters before you begin.

9.3 SNAPPlus2 Configuration Tool

All SNAPPlus2 product configuration is done using the `xsnapadmin` program. This tool is an X-Windows application which provides a graphical interface so that you can view and modify the current SNAPPlus2 configuration and the current running state of the host SNA node. Refer to the HP-UX SNAPPlus2 administrators guide for more details on using `xsnapadmin`.

9.4 Creating SNAPPlus2 Profiles for the Gateway

The Oracle Transparent Gateway for IBM DRDA requires a stored set of definitions, called Side Information Profiles, to support connections between the gateway and DRDA Servers. Each profile consists of a profile name and a profile type, which is a set of fields describing the profile. The fields in a given profile type are generally a mixture of operating parameter values and names of other SNA profiles relevant to the profile. Each functional part of APPC, such as the Mode, Remote Transaction Program name, and Logical Unit (LU), is described by a distinct profile type.

9.5 Independent Versus Dependent LUs

The Gateway configuration can accommodate either independent LUs or dependent LUs. If you choose to use dependent LUs, or are restricted to using dependent LUs, the Gateway will function properly; if a dependent LU is correctly defined, then you will need to make no alterations to the configuration of the Oracle Transparent Gateway for IBM DRDA, nor should any changes be needed to the DRDA Server. However, Oracle Corporation recommends using independent LUs for the Oracle Transparent Gateway for IBM DRDA because they support multiple parallel sessions or conversations. This means that multiple Oracle client applications can be active simultaneously with the same DRDA Server through the independent LU.

In contrast to independent LUs, dependent LUs support only a single active session. The CP (Control Point for the Node) queues each additional conversation request from the gateway behind an already active conversation. In other words, conversations are single-threaded for dependent LUs.

The operational impact of dependent LUs is that the first client application can initiate a conversation through the gateway with the DRDA Server, but while that session is active (which could be seconds, minutes or hours, depending on how the client application and transaction are designed), any other client application initiating a session with the same DRDA Server appears to hang as it waits behind the previous session.

If a production application really uses only a single conversation at any one time, then there should be no problem. However, at some point you might require additional concurrent conversations for testing or for other application development. Having more than one conversation requires that additional dependent LUs be defined on the remote host. Additional configuration entries will need to be added to SNAPPlus2. Additional Side Information Profiles should be defined to use the new dependent LUs. Oracle Transparent Gateway for IBM DRDA instances should be created and configured to use these new Side Information Profiles.

9.6 Creating SNA Definitions for the Gateway

SNAPPlus2 definitions are stored in the following two files, located in the directory `/etc/opt/sna`:

- `sna_node.cfg` - SNA node definitions
- `sna_domn.cfg` - SNA domain definitions

These files are created and maintained with the `xsnapadmin` tool. Maintenance of SNA definitions is normally done by a user with administrative authority. The following information is intended for the person creating SNA definitions for the gateway. You should have some knowledge of SNA before reading this section.

9.6.1 Sample SNAPPlus2 Definitions

The `$ORACLE_HOME/tg4drda/sna/snapplus` subdirectory contains a set of sample SNAPPlus2 definitions for the gateway, created with the `xsnapadmin`. SNA definitions are very specific to the HP9000 host and SNA network. As such, the sample definitions that are provided will not work without being tailored for the local host and SNA network.

9.6.2 Configuring SNAPPlus2

This section describes the process of creating your SNA definitions for SNAPPlus2, using `xsnapadmin`. All of the tasks described in this section are performed from within `xsnapadmin`.

All configuration is done using the various pull-down menus and panels in `xsnapadmin`. The following configuration descriptions follow the samples provided. Please tailor the various SNA values for your local host and SNA network.

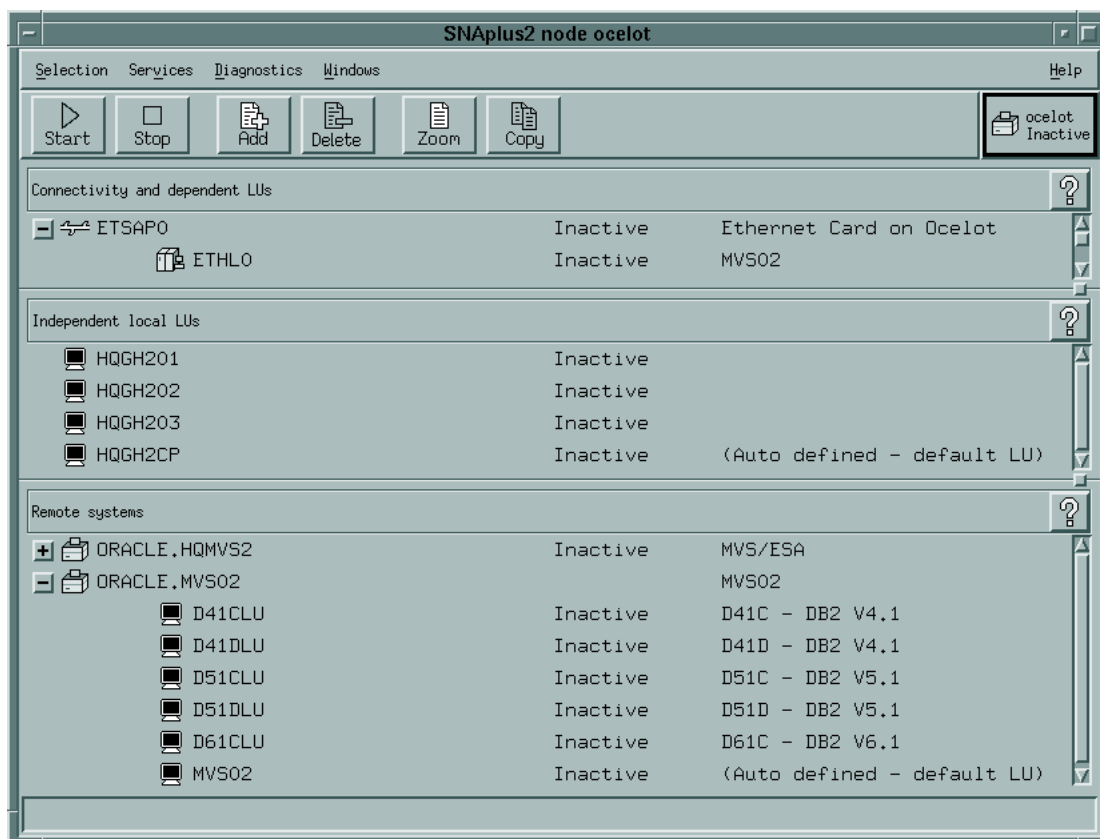
9.6.3 Invoking xsnapadmin

Use the following commands to invoke `xsnapadmin`. The `$DISPLAY` environmental variable must be set appropriately. If you are running `xsnapadmin` from the local HP9000 console, then `$DISPLAY` should already be set. If you are running `xsnapadmin` from a remote X display, then set `$DISPLAY` to the host name or IP address of that display.

```
$ DISPLAY=xstation10.us.oracle.com:0
$ export DISPLAY
$ xsnapadmin &
```

Upon startup of `xsnapadmin`, the main screen will open and display the current configuration of the local SNA node. (See [Figure 9-1](#))

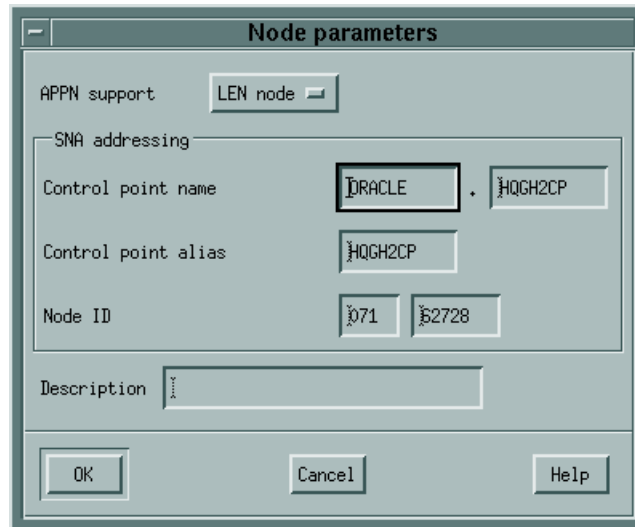
Figure 9-1 *xsnapadmin Main Screen*



Configuring the SNA node

From the Services menu, select Configure Node Parameters. In the Node Parameters dialog box (see [Figure 9-2](#)), enter the APPN support type, Control Point Name, Control Point Alias, and Node ID as needed. The Control Point Name is composed of the SNA Network Name and the CP name of the local host. Click **OK**.

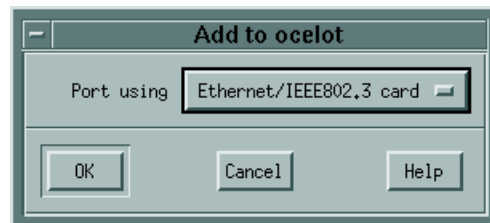
Figure 9-2 Node Parameters Dialog Box



Adding a Port

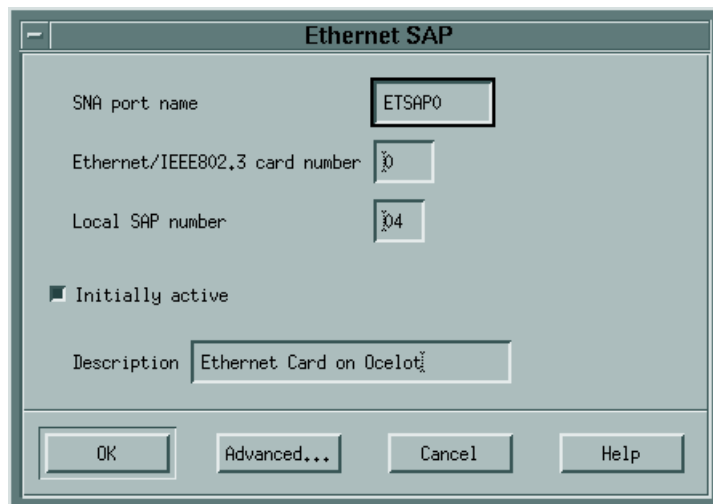
From the Services menu, select Connectivity and Add Port. In the Add to <nodename> dialog box ([Figure 9-3](#)), select the Port type and click **OK**.

Figure 9-3 Add to <nodename> Dialog Box



In the SAP dialog box (see [Figure 9-4](#)), enter a Port name and network card number. The Port name will be used to logically name the physical network card that you are using and will be used to bind a Service Access Port to the card for SNA protocols. Usually, you can accept the values provided in the dialog box. If a different network card is needed, however, enter the card number as reported with the `lanscan` command. Click **OK**.

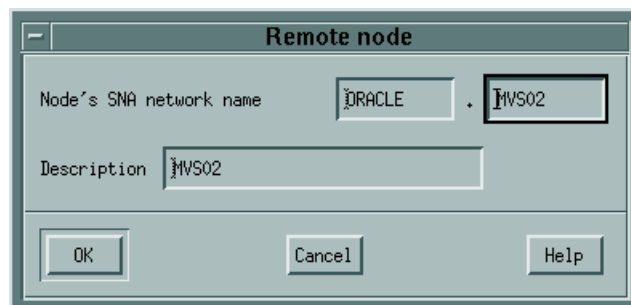
Figure 9-4 Ethernet SAP Dialog Box



Create a Link Station

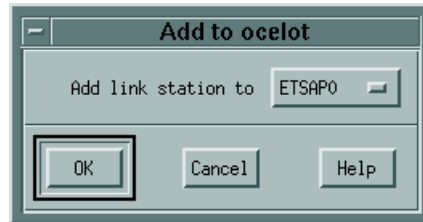
Once the Port has been defined, you need to create a Link Station. The Link Station represents the SNA node of the remote host of the DRDA Server. But before you can create the Link Station, you must create a Remote Node definition. From the Services menu select APPC and Add Remote Node. In the dialog box (see [Figure 9-5](#)) enter the SNA CPNAME of the remote node and click [OK].

Figure 9-5 Add Remote Node Dialog Box



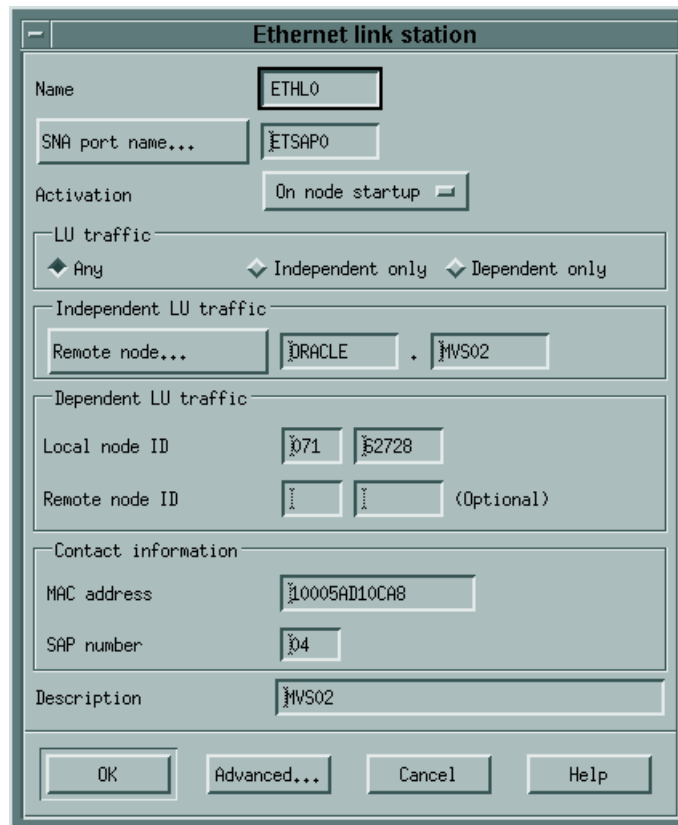
Now you are ready to create the Link Station. From the Services menu, select Connectivity and Add Link Station. In the dialog box (see [Figure 9-6](#)) select the Port previously defined and click [OK].

Figure 9-6 Add Link Station Dialog Box



In the Link Station dialog box (see [Figure 9-7](#)) enter a name for the Link Station, choose the SNA Port name and choose the type of link activation. Choose the LU Traffic type. For maximum flexibility, choose the Any option. For Independent LU traffic, specify the Remote Node name. Click on [Remote Node] and select the node you previously created. Click [OK]. For Dependent LU traffic, specify the Local Node ID, and optionally, Remote Node ID. Then specify the Contact Information.

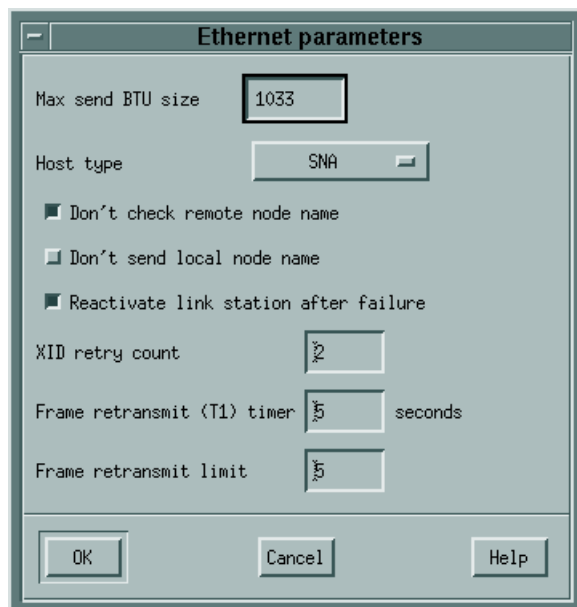
Figure 9-7 Link Station Dialog Box



Contact information contains the MAC address of the remote host as well as the SAP number. Press the [Advanced] button for additional parameters of the Link Station.

The Ethernet Parameters dialog box shows additional parameters of the Link Station (see [Figure 9–8](#)). These parameters effect initial XID contact and retransmission times and limits. The defaults are normally sufficient. Click [OK].

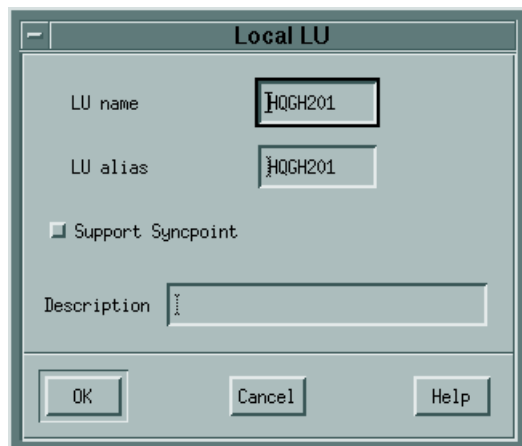
Figure 9–8 Ethernet Parameters Dialog Box



Create Local LUs

Once the Remote Node definitions have been made, create the Local LU names for the local host. From the Services menu select APPC and Add Local LU. In the dialog box (see [Figure 9–9](#)) enter the name of the local LU and an alias. This name must correspond to the VTAM definitions on the remote DRDA Server host for the HP9000 host. Click [OK].

Figure 9–9 Local LU Dialog Box



Create Partner LUs

Now define a Partner LU which represents the LU that the DRDA Server is using to communicate. From the Services menu select APPC and Add Partner LUs and Partner LU on Remote Node. In the dialog box (see [Figure 9–10](#)) Enter the Partner LU name and characteristics. The Partner LU name will contain the SNA Network Name as well as the LU name of the remote LU. Enable parallel session support. The location is the name as the Remote Node name. You may click on [Location] for a list. Click [OK].

Figure 9–10 *Partner LU Dialog Box*



The image shows a dialog box titled "Partner LU". It contains several input fields and a checkbox. The "Partner LU name" field is split into two parts: "DRACLE" and "D41CLU". The "Partner LU characteristics" section includes "Alias" and "Uninterpreted name", both set to "D41CLU" and marked as optional. A checkbox labeled "Supports parallel sessions" is checked. The "Location ..." field is split into "DRACLE" and "MVS02". The "Description" field contains "D41C - DB2 V4.1". At the bottom are "OK", "Cancel", and "Help" buttons.

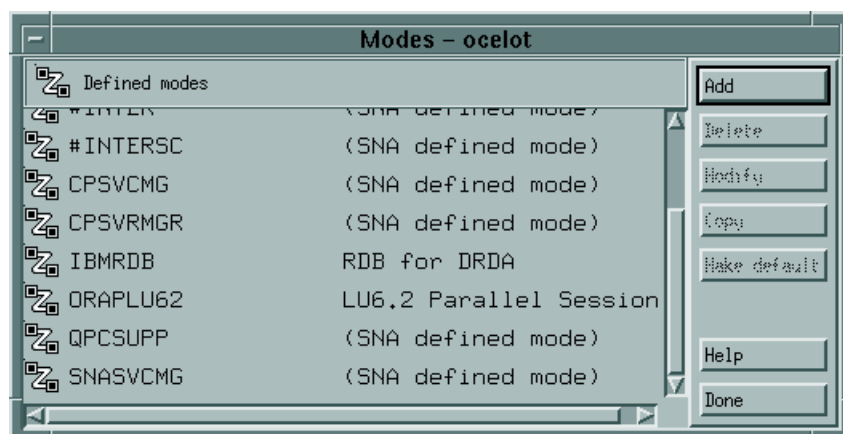
Partner LU	
Partner LU name	DRACLE . D41CLU
Partner LU characteristics	
Alias	D41CLU <Optional>
Uninterpreted name	D41CLU <Optional>
<input checked="" type="checkbox"/> Supports parallel sessions	
Location ...	DRACLE . MVS02
Description	D41C - DB2 V4.1
OK Cancel Help	

Create Mode and CPI-C Profiles

Once the local and remote LU definitions have been made, create the necessary Mode and CPI-C definitions.

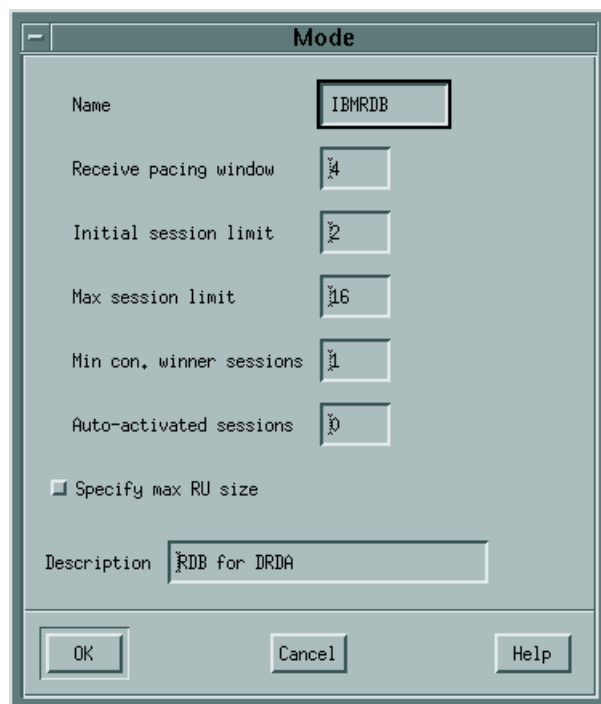
From the Services menu select APPC and Modes. In the Modes dialog box (see [Figure 9-11](#)) click on Add to add a new mode.

Figure 9-11 Modes Dialog Box



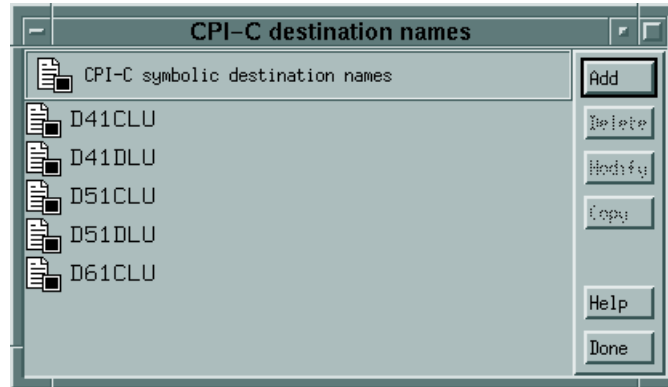
In the Mode dialog box (see [Figure 9-12](#)) enter the Mode Name and other session parameters. The prescribed name for a DRDA mode is "IBMRDB". Contact your Remote Host system administrator for appropriate mode parameters. Click **OK**.

Figure 9-12 IBMRDB Mode Dialog Box



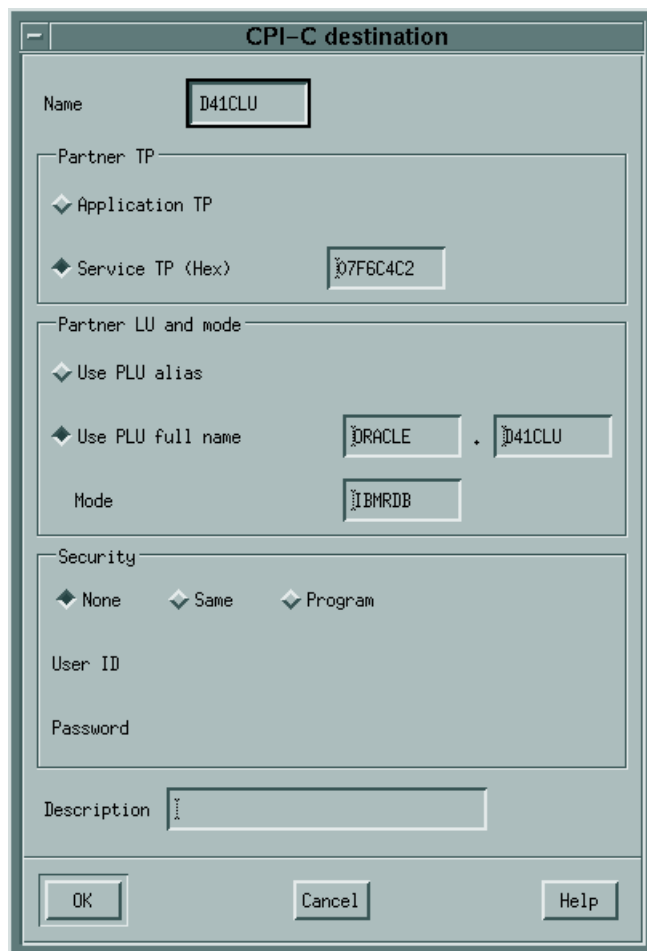
Now that the Mode has been defined, create the CPI-C Side Information Profile, which the gateway will use as a connection name. From the menu select APPC and CPI-C. In the CPI-C destination names dialog box (see [Figure 9-13](#)) click on Add to add a new Profile.

Figure 9-13 *CPI-C Destination Names Dialog Box*



In the CPI-C destination dialog box (see [Figure 9–14](#)) enter the Profile name, Partner TP, Partner LU, mode and Security option. The default TP name of the mode DRDA Server will typically be a Service TP named "07F6C4C2". For the Partner LU, enter either the full LU name or the alias created previously. Enter "IBMRDB" for the mode name. Lastly, choose the type of security these sessions will use. This will affect how session authorization is done. Click [OK].

Figure 9–14 CPI-C Destination Dialog Box



The image shows a dialog box titled "CPI-C destination". It contains several sections for configuring a destination:

- Name:** A text field containing "D41CLU".
- Partner TP:** A section with two radio buttons: "Application TP" (unselected) and "Service TP (Hex)" (selected). The "Service TP (Hex)" option has a text field containing "07F6C4C2".
- Partner LU and mode:** A section with two radio buttons: "Use PLU alias" (unselected) and "Use PLU full name" (selected). The "Use PLU full name" option has two text fields: "ORACLE" and "D41CLU", separated by a dot. Below these is a "Mode" text field containing "IBMRDB".
- Security:** A section with three radio buttons: "None" (selected), "Same" (unselected), and "Program" (unselected). Below these are "User ID" and "Password" text fields.
- Description:** A text area at the bottom.
- Buttons:** "OK", "Cancel", and "Help" buttons at the bottom.

9.7 Using SNA Session Security Validation

When the database link request for the gateway begins, the gateway attempts to start an APPC conversation with the DRDA Server. Before the conversation can begin, a session must start between the HP9000 host Logical Unit (LU) and the DRDA Server LU.

SNA and its various access method implementations (including SNAPPlus2 and VTAM) provide security validation at session initiation time, allowing each LU to authenticate its partner. This is carried out entirely by network software before the gateway and server application programs begin their conversation and process conversation-level security data. If session-level security is used, then correct password information must be established in the HP9000 host Connection Profile and in similar parameter structures in the DRDA Server system that is to be accessed. Refer to the appropriate SNA server product documentation for detailed information.

9.8 SNA Conversation Security

SNA conversation security is determined by the setting of the gateway initialization parameter, `DRDA_SECURITY_TYPE`. This parameter determines whether SNA security option `SECURITY` is set to `PROGRAM` or to `SAME`. Generally, the gateway operates under SNA option `SECURITY=PROGRAM`, but it can also be set to operate under SNA option `SECURITY=SAME`.

9.8.1 SNA Security Option `SECURITY=PROGRAM`

If `DRDA_SECURITY_TYPE=PROGRAM` is specified, then the gateway allocates the conversation with SNA option `SECURITY=PROGRAM` and sends this information to the user ID:

- If the database link has explicit `CONNECT` information, then the specified user ID and password are sent.
- If the database link has no `CONNECT` clause and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID and password are sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID and password are sent. If no user ID and password are sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

In general, `SECURITY=PROGRAM` tells the DRDA Server to authenticate the user ID/password combination using whatever authentication mechanisms are available. For example, if DB2/OS390 is the DRDA Server, then RACF can be used. This is not always the case, however, because each of the DRDA Servers can be configured to process inbound user IDs in other ways.

9.8.2 SNA Security Option SECURITY=SAME

If `DRDA_SECURITY_TYPE=SAME` is specified, then the gateway allocates the conversation with SNA option `SECURITY=SAME`, and the following information is sent to the DRDA Server:

- If the database link has explicit `CONNECT` information, then the specified user ID is sent.
- If the database link has no `CONNECT` clause, and if the application has logged into the Oracle integrating server with an explicit user ID and password, then the Oracle user ID is sent.
- If the application logs into the Oracle integrating server with operating system authentication, and if the database link lacks explicit `CONNECT` information, then no user ID is sent. If no user ID is sent, and if the DRDA Server is not configured to assign a default user ID, then the connection fails.

For this option to function properly, SNAPPlus2 requires that the effective user ID under which the gateway is executing must be a member of the system group. In UNIX terms, this means that the user ID must be defined with its primary group set to `system`. In addition, the owning user ID of the gateway executable must be set to the desired effective user ID, and the set-uid bit of the executable file permissions must also be set. The `ls -l` command shows the owning user ID and the setting of the set-uid bit for the executable file. The owning user ID can be changed by the root user with the `chown` command, and the set-uid bit can be set using the `chmod u+s` command. The gateway executable, as installed by the Oracle Universal Installer, has its set-uid bit disabled.

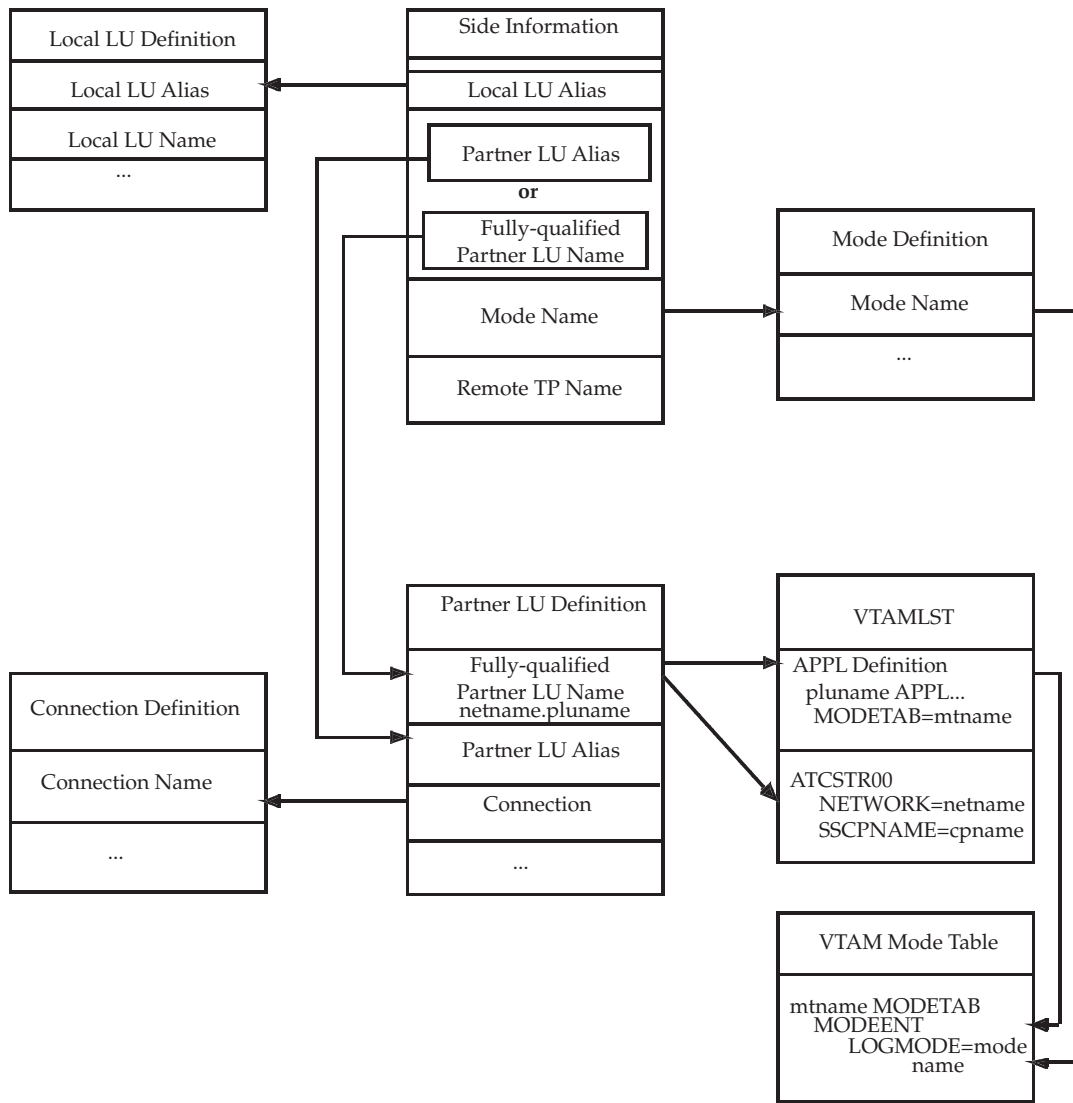
The simplest way to cause the gateway to execute under an effective user ID that is a member of the system group is to change the owning user ID of the gateway executable to `root`. Another way is to change the primary group for the owning user ID of the gateway executable to `system`. However, be careful when choosing the user ID. Oracle Corporation recommends using `root` and recommends never changing the Oracle dba user ID primary group to `system`.

When the effective user ID is not a member of the system group, a failure is generated when the gateway attempts to allocate a conversation with the DRDA Server, and an error message is sent to the gateway user.

9.9 Testing the Connection

Before proceeding with the gateway configuration tasks in [Chapter 12, "Configuring the Gateway"](#), ensure that your connection is working. Do this by starting the SNAPPlus2 Node and then starting the individual link stations.

[Figure 9-15](#) shows the relationship between SNAPPlus2 definitions and the VTAM definitions on the remote host.

Figure 9–15 Relationship between SNAPPlus2 Definitions and Host VTAM Definitions

Configuring TCP/IP

This chapter describes configuring TCP/IP for the various Unix platforms supported by the Oracle Transparent Gateway for DRDA. TCP/IP is a communication facility that is already part of the operating system. No third-party protocol software is required. Read this chapter to learn more about configuring TCP/IP.

This chapter contains the following sections:

- [Before You Begin](#) on page 10-2
- [Configuring TCP/IP under UNIX](#) on page 10-2

10.1 Before You Begin

This chapter requires you to enter parameters that are unique to your system in order to properly configure TCP/IP. Refer to [Appendix E](#) for a worksheet listing all of the installation parameters that you will need to know about before you can complete the configuration process. Ask your network administrator to provide you with these parameters before you begin.

10.2 Configuring TCP/IP under UNIX

Basic configuration consists of assigning a Hostname, an IP Address, and a Network Mask to a given network interface. This basic configuration should have been completed already by the System Administrator. If not, contact your System Administrator to have this configuration completed before you continue.

Additional configuration consists of defining a Name Server IP Address or creating entries in the Hosts file on the local machine. Name Servers translate hostnames into IP Addresses when queried on a particular host name. The Hosts file provides this same functionality, but in a non-network participating manner.

For local configuration (i.e., the gateway and the DRDA Server are on the same machine), it may be desirable to use the loop-back address. The IP address is 127.0.0.1 and is typically given the local name ("localhost" or "loopback") in the Hosts file. Using the loop-back address reduces the amount of network overhead by handling the traffic internally without actually talking to the network.

The gateway is configured for TCP/IP using the DRDA_CONNECT_PARM initialization file parameter. In an SNA configuration, this parameter would be set to the Side Information Profile name. In a TCP/IP configuration, this parameter should be set to the IP address or Host name of the DRDA Server, which should be followed by the Service Port number of that server.

Note: When installing the gateway, you must choose either SNA or TCP/IP for the Networking Interface.

The DRDA_CONNECT_PARM must be configured correctly for the chosen Networking Interface.

The rest of the DRDA-specific parameters are unrelated to the communications protocol and may be set the same for either SNA or TCP/IP installations, as illustrated below.

Example 1

Configuration for a DRDA Server on a host named 'mvs01.domain.com' (or IP address of 192.168.1.2) with a Service Port number of 446.

```
DRDA_CONNECT_PARM=mvs01.domain.com:446
```

or

```
DRDA_CONNECT_PARM=192.168.1.2:446
```

Example 2

Configuration for a DRDA Server on the same host as the gateway with a Service Port number of 446.

```
DRDA_CONNECT_PARM=localhost:446
```

or

```
DRDA_CONNECT_PARM=127.0.0.1:446
```

For additional information about configuring TCP/IP for a particular host operating system, refer to the appropriate platform and operating system installation and configuration guides.

Oracle Net is an Oracle product providing network communication between Oracle applications, Oracle Database servers, and Oracle Gateways across different systems

This chapter contains the following sections:

- [Checklists for Oracle Net](#) on page 11-2
- [Oracle Net Introduction](#) on page 11-3
- [Oracle Net Overview](#) on page 11-3
- [Configuring Oracle Net](#) on page 11-4
- [Advanced Security Encryption](#) on page 11-5
- [Setting Up Advanced Security Encryption for Test](#) on page 11-6
- [Testing Advanced Security Encryptions](#) on page 11-6

11.1 Checklists for Oracle Net

Use the following checklists when you are installing and configuring Oracle Net.

11.1.1 Configuring Oracle Net

- ❑ Step 1: [Modify listener.ora file](#)
- ❑ Step 2: [Modify tnsnames.ora file](#)

11.1.2 Advanced Security Encryption

Verifying if CHECKSUM and the Export encryption algorithms are used at your site:

11.1.2.1 Setting Up Advanced Security Encryption for Test

- ❑ Step 1: [Set Advanced Security Encryption Parameters for the Gateway](#)
- ❑ Step 2: [Set Advanced Security Encryption Parameters for Oracle Server](#)

11.1.2.2 Testing Advanced Security Encryptions

- ❑ Step 1: [Connect Gateway and Oracle Integrating Server](#)
- ❑ Step 2: [Reset Configuration Parameters on the Gateway](#)

11.2 Oracle Net Introduction

Oracle Net provides connectivity to the Gateway through the use of Protocol Adapters, SQL*Net, and the TNS Listener. Configuration of Oracle Net is backwards compatible with past versions of SQL*Net. A new facility called Heterogeneous Services (HS) has been added to both Oracle Net and the Gateway to improve the throughput of data. For additional information, refer to *Oracle Net Services Administrator's Guide*, *Oracle Net Services Reference Guide*, and *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

11.3 Oracle Net Overview

Oracle Net is a required Oracle product supporting network communications between Oracle applications, Oracle Database servers, and Oracle gateways across different CPUs or operating systems. It also supports communication across different Oracle Databases and CPUs providing distributed database and distributed processing capabilities.

Oracle Net also allows applications to connect to multiple Oracle Database servers or gateways across a network, selecting from a variety of communications protocols and application program interfaces (APIs) to establish a distributed processing and distributed database environment.

A communications protocol is a set of implemented standards or rules governing data transmission across a network. An API is a set of subroutines providing an interface for application processes to the network environment.

11.3.1 Distributed Processing.

Dividing processing between a front-end computer running an application and a back-end computer used by the application is known as distributed processing. Oracle Net enables an Oracle tool or application to connect to a remote computer containing an Oracle Database server or Oracle gateway.

11.3.2 Distributed Database

Several databases linked through a network, appearing as a single logical database, are known as a distributed database. An Oracle tool running on a client computer or on an Oracle Database server running on a host computer can share and obtain information retrieved from other remote Oracle Database servers. Regardless of the number of database information sources, you might be aware of only one logical database.

11.3.3 Terminology for Oracle Net

The following terms are used to explain the architecture of Oracle Net:

host is the computer the database resides on and that runs the Oracle Database server or gateway.

client (task) is the application using a Oracle Net driver to communicate with the Oracle Database server or gateway.

protocol is a set of standards or rules governing the operation of a communication link.

driver is the part of Oracle Net supporting a given network protocol or communication method.

network is a configuration of devices and software connected for information interchange.

11.4 Configuring Oracle Net

The gateway must be defined to the TNS listener, and a service name must be defined for accessing the gateway.

11.4.1 Step 1: Modify listener.ora file

Add an entry for the gateway to the **listener.ora** file. For example:

```
(SID_DESC=
  (SID_NAME=sidname)
  (ORACLE_HOME=/oracle/tg4drda/10.1.0)
  (PROGRAM=g4drsrv))
```

Refer to [Appendix B, "Sample Files"](#), for a sample **listener.ora** file.

Note: The `PROGRAM=g4drsrv` parameter is required. It specifies to the listener the name of the gateway executable.

11.4.2 Step 2: Modify tnsnames.ora file

Add a gateway service name to the **tnsnames.ora** file on the system where your Oracle integrating server resides. Specify the service name in the USING parameter of the database link defined for accessing the gateway from the Oracle Database 10g server.

You can use the IPC protocol only if the Oracle integrating server and the gateway reside on the same machine. If you use the IPC protocol adapter, then add an entry like this to **tnsnames.ora**:

```
linkname1 = (DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=IPC)
                (KEY=ORAIPC) )
              (CONNECT_DATA=(SID=sidname))
              (HS=)
            )
```

where:

linkname1 is the name used to define the database link referencing the gateway.

ORAIPC is the IPC key defined in the **listener.ora** file for the IPC protocol.

sidname is your gateway SID, the same SID that you used for the entry in your **listener.ora** file.

If you are using the TCP/IP protocol adapter, then add this entry to **tnsnames.ora**:

```
linkname2 = (DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=port)
                (HOST=hostname) )
              (CONNECT_DATA=(SID=sidname))
              (HS=)
            )
```

where:

linkname2 is the name used to define the database link referencing the gateway.

port is the TCP/IP port number on which the Oracle listener is listening (default is 1541).

hostname is the name of your host system.

sidname is your gateway SID.

Refer to ["Sample Oracle Net tnsnames.ora File"](#) on page B-3 for a sample **tnsnames.ora** file. For more information about configuring Oracle Net, refer to *Oracle Net Services Administrator's Guide*.

11.5 Advanced Security Encryption

Oracle Net supports the CHECKSUM command and the Export encryption algorithms. The following sections describe a basic method of verifying this feature if it is used at your site. The easiest way to determine if Advanced Security encryption is attempting to work is to deliberately set wrong configuration parameters and attempt a connection between the server and client. Incorrect parameters cause the connection to fail.

After receiving the expected failure message, set the configuration parameters to the correct settings and try the connection again. Encryption is working properly if you receive no further error messages.

11.6 Setting Up Advanced Security Encryption for Test

The following procedures test Advanced Security encryption by the above method. The incorrect parameter settings produce error 12660

1. Set Advanced Security encryption parameters for the gateway
2. Set Advanced Security encryption parameters for the Oracle integrating server

Note: The international or export version of Advanced Security encryption supports the following encryption types:

- des40
 - rc4_40
-

11.6.1 Step 1: Set Advanced Security Encryption Parameters for the Gateway

Edit the Oracle Net configuration file on the host system (gateway system) to add the following parameters and values:

```
SQLNET.CRYPTO_CHECKSUM_SERVER = REJECTED
SQLNET.ENCRYPTION_SERVER = REJECTED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (MD5)
SQLNET.ENCRYPTION_TYPES_SERVER = (DES40,RC4_40)
SQLNET.CRYPTO_SEED = "abcdefgh123456789"
```

The value shown for SQLNET.CRYPTO_SEED is only an example. Set it to the value you want. Refer to the *Advanced Security Administrator's Guide* for more information.

11.6.2 Step 2: Set Advanced Security Encryption Parameters for Oracle Server

Set the advanced security encryption parameters for the Oracle integrating server.

Edit the Oracle Net configuration file on the Oracle integrating server system to add the following parameters:

```
SQLNET.CRYPTO_CHECKSUM_CLIENT = REQUIRED
SQLNET.ENCRYPTION_CLIENT = REQUIRED
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (MD5)
SQLNET.ENCRYPTION_TYPES_CLIENT = (DES40,RC4_40)
SQLNET.CRYPTO_SEED = "abcdefgh123456789"
```

The value shown for SQLNET.CRYPTO_SEED is only an example.

11.7 Testing Advanced Security Encryptions

After completing Steps 1 and 2 to set up Advanced Security encryption, you are ready to test the operation of the Advanced Security encryption by using the following steps:

1. Connect gateway and Oracle integrating server
2. Reset configuration parameters on the gateway

11.7.1 Step 1: Connect Gateway and Oracle Integrating Server

Use SQL*Plus to logon to the Oracle integrating server. Access the gateway through a database link. You should receive the following error:

```
ORA-12660: Encryption or crypto-checksumming
```

11.7.2 Step 2: Reset Configuration Parameters on the Gateway

Change the following Advanced Security encryption parameters on the gateway to:

```
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUIRED
```

```
SQLNET.ENCRYPTION_SERVER = REQUIRED
```

Attempt the connection between the gateway and the Oracle integrating server again.

If no error message is returned and the connection completes, then you can assume Advanced Security encryption is working properly.

Configuring the Gateway

After you have installed the gateway, configured your DRDA Server, and configured your SNA or TCP/IP software, then you must configure the gateway. Some of these tasks involve customizing the Gateway Initialization File.

This chapter includes the following sections:

- [Configuration Checklists](#) on page 12-2
- [Choosing a Gateway System Identifier \(SID\)](#) on page 12-4
- [Gateway Configuration](#) on page 12-4
- [Configuring the Host](#) on page 12-4
- [DRDA Gateway Package Considerations](#) on page 12-7
- [Backup and Recovery of Gateway Configuration](#) on page 12-8
- [Configuring the Oracle Integrating Server](#) on page 12-8
- [Accessing the Gateway from Other Oracle Database Servers](#) on page 12-9
- [Accessing Other DRDA Servers](#) on page 12-9
- [Gateway Installation and Configuration Complete](#) on page 12-9

12.1 Configuration Checklists

Configuring the Gateway

- ❑ [Choosing a Gateway System Identifier \(SID\)](#)

Configuring the Host

- ❑ [Step 1: Choose the initsid.ora file](#)
- ❑ [Step 2: Tailor the initsid.ora file](#)

Binding the DRDA Gateway Package

- ❑ [Step 1: Log on to an Oracle integrating server.](#)
- ❑ [Step 2: Create a Database link.](#)
- ❑ [Step 3: Execute the stored procedure GTW\\$_BIND_PKG:](#)

Binding Packages on DB2/Universal Database (DB2/UDB)

- ❑ [Step 1: Log into the machine where DB2/UDB is running.](#)
- ❑ [Step 2: Copy files from \\$ORACLE_HOME/tg4drda/install/db2udb.](#)
- ❑ [Step 3: Connect to the database.](#)
- ❑ [Step 4: Create the ORACLE2PC table:](#)
- ❑ [Step 5: Commit the transaction:](#)
- ❑ [Step 6: Verify that the table was created.](#)
- ❑ [Step 7: Disconnect from the session:](#)

Before Binding the DRDA Gateway Package

- ❑ [Step 1: Check all DRDA parameter settings](#)
- ❑ [Step 2: If using DB2/UDB, then create ORACLE2PC table](#)

Sample SQL scripts

- ❑ [Step 1: If server is DB2/OS390, DB2/400, or DB2/UDB, then run data dictionary scripts](#)
- ❑ [Step 1a: Upgrading from a previous gateway version](#)
- ❑ [Step 1b: Creating the Data Dictionary tables and views](#)
- ❑ [Step 2: DB2/UDB or other server](#)
- ❑ [Step 2a: If server is DB2/UDB, grant authority to package](#)
- ❑ [Step 2b: If server is not DB2/UDB, create the ORACLE2PC table](#)

Configuring the Oracle Integrating Server

- ❑ [Step 1: Create a database link](#)
- ❑ [Step 2: Create synonyms and views](#)

Accessing the Gateway from Other Oracle Database Servers

- ❑ Step 1: Create a database link with which to access the gateway.
- ❑ Step 2: If needed, define synonyms and views for tables accessed through the gateway.
- ❑ Step 3: Perform GRANT statements for the synonyms and views you create.

Accessing Other DRDA Servers

- ❑ Step 1: Configure another SNA profile set for the DRDA Server.
- ❑ Step 2: Configure additional DRDA Server instances.
- ❑ Step 3: Bind the DRDA package to your DRDA Server.

12.2 Choosing a Gateway System Identifier (SID)

The gateway SID is a string of alphabetic and numeric characters that identifies a gateway instance. The SID is used in the filenames of gateway parameter files and in the connection information associated with the Oracle Database server database links that access the gateway.

A separate SID is required for each DRDA Server to be accessed. You might also have multiple SIDs for one DRDA Server to use different gateway parameter settings with that server. Refer to "[Accessing Other DRDA Servers](#)" on page 12-9 for information on configuring additional SIDs.

12.3 Gateway Configuration

The information in this chapter describes the configuration process for the gateway. All gateway parameters are kept in the **initsid.ora** gateway initialization file, which is stored in the gateway **admin/** directory.

12.4 Configuring the Host

To configure the host for the Oracle Transparent Gateway for DRDA, you tailor the parameter files for your installation.

12.4.1 Step 1: Choose the **initsid.ora** file

The **initsid.ora** gateway initialization file defines the operating parameters for the gateway. Samples (tailored for each type of DRDA Server) are provided as a starting point for tailoring to your particular installation. The samples are stored in the **\$ORACLE_HOME/tg4drda/admin** directory. The following is a list of the initialization files for various DRDA Server platforms:

- sample initialization file for DB2/OS390: **initDB2.ora**
- sample initialization file for DB2/UDB: **initDB2UDB.ora**
- sample initialization file for DB2/400: **initAS400.ora**
- sample initialization file for DB2/VM: **initDB2VM.ora**

Choose a sample initialization file and copy it, within the same directory, to the name of the gateway SID, using the following naming convention:

`initSID.ora`

where *SID* is the chosen gateway SID. For example, if the chosen gateway SID were DRDA, then the initialization file would be named `initDRDA.ora`.

12.4.2 Step 2: Tailor the *initsid.ora* file

After you have copied the sample initialization file, you will need to tailor it to your installation. While many parameters can be left to their defaults, some parameters must be changed for correct operation of the gateway. Attention should be given to the following DRDA and HS parameters. Attention should also be given to the security aspects of the initialization file. [Chapter 15, "Security Considerations"](#), contains details on using the **g4drpwd** utility to handle encryption of passwords that would otherwise be embedded in the initialization file. See [Appendix C, "DRDA-Specific Parameters"](#), for a description of the following parameters:

- DRDA_CONNECT_PARM
- DRDA_PACKAGE_COLLID
- DRDA_PACKAGE_NAME
- DRDA_PACKAGE_OWNER
- DRDA_REMOTE_DB_NAME
- HS_DB_NAME
- HS_DB_DOMAIN
- FDS_CLASS

12.4.3 Binding the DRDA Gateway Package

The product requires a package to be bound on the DRDA Server. The gateway has an internal, stored procedure that must be used to create this package. The internal, stored procedure is invoked from an Oracle integrating server. (Refer to ["Configuring Oracle Net"](#) on page 11-4. Also refer to ["Configuring the Oracle Integrating Server"](#) in this chapter on page 12-8.) Before this package can be bound on the DRDA Server, the Gateway Initialization File must be correctly configured (refer to [Appendix C, "DRDA-Specific Parameters"](#)).

1. Log on to an Oracle integrating server.

Use either SQL*Plus or Server Manager:

```
$ sqlplus system/manager
```

2. Create a Database link.

Create a Database link with a user ID and with a password that has proper authority on the DRDA Server to create packages.

```
SQL> CREATE PUBLIC DATABASE LINK dblink
2 CONNECT TO userid IDENTIFIED BY password
3 USING 'tns_name_entry'
```

Note: The user ID that is creating the public database link must have the "CREATE PUBLIC DATABASE LINK" privilege.

Refer to ["Configuring the Oracle Integrating Server"](#) on page 12-8 for more information.

3. Execute the stored procedure GTW\$_BIND_PKG:

```
SQL> exec GTW$_BIND_PKG@dblink;
SQL> COMMIT;
```

This creates and commits the package. If any errors are reported, then correct the Gateway Initialization File parameters as needed and re-execute the bind procedure above.

12.4.4 Binding Packages on DB2/Universal Database (DB2/UDB)

If you are connecting to a DB2/UDB DRDA Server, then DB2/UDB requires that you create the **ORACLE2PC** table before binding the DRDA package. Other DRDA Servers allow you to bind the package before the ORACLE2PC table exists.

To create the ORACLE2PC table:

1. Log into the machine where DB2/UDB is running.
Check that you have the ability to address the DB2/UDB instance where the ORACLE2PC table will reside.
2. Copy files from **\$ORACLE_HOME/tg4drda/install/db2udb**.
Copy the following files from the **\$ORACLE_HOME/tg4drda/install/db2udb** directory:
 - **o2pc.sh** (Sample shell script for performing the table creation)
 - **o2pc.sql** (SQL script for creating the table)
 - **o2pcg.sql** (SQL script for granting package access to PUBLIC)

3. Connect to the database.

Connect to the database using the user ID that you will use for binding the package:

```
$ db2 'CONNECT TO database USER userid USING password'
```

Note: The user ID must have CONNECT, CREATETAB, and BINDADD authority to be able to connect to the database, create the table, and create the package.

For more information, refer to ["DB2/UDB \(Universal Database\)"](#) on page 5-5.

4. Create the ORACLE2PC table:

```
$ db2 -tf o2pc.sql
```

5. Commit the transaction:

```
$ db2 'COMMIT'
```

6. Verify that the table was created.

Optionally, verify the table was created under the correct user ID:

```
$ db2 'LIST TABLES FOR USER'
```

```
$ db2 'COMMIT'
```

7. Disconnect from the session:

```
$ db2 'DISCONNECT CURRENT'
```

12.5 DRDA Gateway Package Considerations

The DRDA package must be bound with the internal Stored Procedure GTW\$_BIND_PKG. You must perform this bind step if this release is the first time the gateway has been installed on this system. If you are upgrading from version 9 of the gateway, then a rebind is not necessary unless the initialization parameters have been changed.

The user ID used to bind or rebind the DRDA package must have the appropriate privileges on the remote database, as described in [Chapter 5, "Configuring the DRDA Server"](#).

12.5.1 Before Binding the DRDA Gateway Package

Check DRDA parameter settings and create your ORACLE2PC table before binding the DRDA gateway package.

12.5.1.1 Step 1: Check all DRDA parameter settings

Check all DRDA parameter settings to be sure that they are set correctly before you start the bind. For example, the default for DRDA_DISABLE_CALL only works if your DRDA database supports stored procedures. If not, then you must change the setting. Also, the value for DRDA_PACKAGE_NAME must be unique if you have any older versions of the gateway installed. New packages replace any old packages with the same name, causing versions of the gateway that use the old package to fail. Refer to [Appendix C](#) for information on the parameters and their settings.

12.5.1.2 Step 2: If using DB2/UDB, then create ORACLE2PC table

If your DRDA Server is DB2/UDB, then create your ORACLE2PC table. Refer to ["Binding Packages on DB2/Universal Database \(DB2/UDB\)"](#) on page 12-6 for information on creating the table.

12.5.2 Sample SQL scripts

SQL scripts are provided to perform steps such as creating the ORACLE2PC table, removing obsolete tables and views, using previous releases, and creating tables and views to provide Data Dictionary support.

Choose the appropriate subdirectory for your DRDA Server platform from the following list:

- for DB2/OS390: choose **tg4drda/install/db2**
- for DB2/400: choose **tg4drda/install/as400**
- for DB2/VM: choose **tg4drda/install/db2vm**
- for DB2/UDB: choose **tg4drda/install/db2udb**

These scripts must be run on the DRDA Server platform using a database native tool (such as SPUFI on DB2/OS390), because no tool is provided with the gateway to execute these scripts. Note that when running these scripts, the user ID used must be suitably authorized.

12.5.2.1 Step 1: If server is DB2/OS390, DB2/400, or DB2/UDB, then run data dictionary scripts

If your DRDA Server is DB2/OS390, DB2/400, or DB2/UDB, then run the following scripts to create the Data Dictionary tables and view.

Step 1a: Upgrading from a previous gateway version

If you are upgrading from a previous version of the gateway then run the dropold.sql script to drop the old data dictionary definitions:

Step 1b: Creating the Data Dictionary tables and views

Run the `g4ddtab.sql` and `g4ddview.sql` scripts to create the Data Dictionary tables and views:

Step 2: DB2/UDB or other server

Depending on your DRDA Server, perform one of the following steps:

Step 2a: If server is DB2/UDB, grant authority to package

If your DRDA Server is DB2/UDB, then the ORACLE2PC table has already been created (see the previous sections). For all users to be able to use the table, run `o2pcg.sql` granting authority to all users.

Step 2b: If server is not DB2/UDB, create the ORACLE2PC table

If your DRDA Server is not DB2/UDB, then the ORACLE2PC table must be created. Run `o2pc.sql`.

12.6 Backup and Recovery of Gateway Configuration

The configuration of the gateway is stored in the Gateway Initialization File. This file is stored in the `$ORACLE_HOME/tg4drda/admin` directory. The Gateway Initialization File is a simple text file. You may back up this file by using an archiving tool of your choice.

12.7 Configuring the Oracle Integrating Server

Configure the Oracle integrating server, regardless of the platform on which it is installed. It can be on the host, but this is not required.

12.7.1 Step 1: Create a database link

To access the DRDA Server, you must create a public database link. A public database link is the most common of database links. Refer to ["Processing a Database Link"](#) on page 13-2 for information on creating database links. In the following example, the Oracle Database server gateway is on the same host. Replace `linkname` with the name you used for the database link when you added your entry to the `tnsnames.ora` file (refer to [Chapter 11, "Oracle Net"](#), ["Step 2: Modify tnsnames.ora file"](#) on page 11-5).

```
CREATE PUBLIC DATABASE LINK DB2 USING 'tns_name_entry'
```

Note: The user ID creating the public database link must have the "CREATE PUBLIC DATABASE LINK" privilege.

12.7.2 Step 2: Create synonyms and views

To facilitate accessing data using the gateway, define synonyms and views for the DRDA data tables. If needed, perform GRANT statements to ensure that the synonyms and views are accessible to the appropriate groups of users. Refer to ["Using the Synonym Feature"](#) on page 13-4 for information.

12.8 Accessing the Gateway from Other Oracle Database Servers

Perform the following steps for each of the Oracle Database servers from which you want to access the gateway:

1. Create a database link with which to access the gateway.
2. If needed, define synonyms and views for tables accessed through the gateway.
3. Perform GRANT statements for the synonyms and views you create.

Provide local or Oracle Net access from the Oracle Database servers to the gateway.

12.9 Accessing Other DRDA Servers

To access other DRDA Servers from the Oracle integrating server, use the following steps:

1. Configure another SNA profile set for the DRDA Server.

Only Side Information and Partner LU Profiles must be new. You can point to existing configuration information for other profiles, unless you need to modify other aspects of the connection. For example, if you are using a different network adapter, then you must configure an entire SNA profile set. No additional profiles need to be configured for TCP/IP.

2. Configure additional DRDA Server instances.

To configure an additional instance, create new Gateway Initialization Files. If you are using Oracle Net, then add entries to the **listener.ora** file and **tnsnames.ora** file with the new SIDs.

Other components, including the gateway **ORACLE_HOME** directory structure, can be shared among multiple gateway instances.

3. Bind the DRDA package to your DRDA Server.

12.10 Gateway Installation and Configuration Complete

The Oracle Transparent Gateway for DRDA installation and configuration process is now complete. The gateway is ready for use.

Using the Gateway

Using the gateway involves connecting to the gateway system and the remote DRDA database associated with it. It is important to understand how to process and use database links. Database links are discussed in detail in the *Oracle Database Reference*. Read the database link information in that guide to understand database link processing. Then proceed to read this chapter to understand how to set up a database link to a remote DRDA database.

This chapter contains the following sections:

- [Processing a Database Link](#) on page 13-2
- [Accessing the Gateway](#) on page 13-3
- [Accessing AS/400 File Members](#) on page 13-3
- [Using the Synonym Feature](#) on page 13-4
- [Performing Distributed Queries](#) on page 13-4
- [Read-Only Gateway](#) on page 13-5
- [Replicating in a Heterogeneous Environment](#) on page 13-6
- [Copying Data from Oracle Database 10g Server to DRDA Server](#) on page 13-6
- [Copying Data from DRDA Server to Oracle Database 10g Server](#) on page 13-7
- [Tracing SQL Statements](#) on page 13-7

13.1 Processing a Database Link

The database and application administrators of a distributed database system are responsible for managing the necessary database links that define paths to the DRDA database.

13.1.1 Creating Database Links

To create a database link and define a path to a remote database, use the CREATE DATABASE LINK statement. The CONNECT TO clause specifies the remote user ID and password to use when creating a session in the remote database. The USING clause points to a **tnsnames.ora** connect descriptor.

Note: If you do not specify a user ID and a password in the CONNECT TO clause, then the Oracle Database server user ID and password are used. For additional information, refer to [Chapter 15, "Security Considerations"](#).

The following syntax creates a database link to access information in the DRDA Server database:

```
CREATE PUBLIC DATABASE LINK dblink
CONNECT TO userid IDENTIFIED BY password
USING 'tns_name_entry';
```

where:

dblink is the complete database link name.

user id is the user ID used to establish a session in the remote database. This user ID must be a valid DRDA Server user ID. It must be authorized to any table or file on the DRDA Server that is referenced in the SQL commands. The user ID cannot be longer than eight characters.

password is the password used to establish a session in the remote database. This must be a valid DRDA Server password. The password cannot be longer than eight characters.

tns_name_entry specifies the Oracle Net TNS connect descriptor used to identify the gateway.

13.1.2 Guidelines for Database Links

Database links are active for the duration of a gateway session. If you want to close a database link during a session, then use the ALTER session statement.

13.1.3 Dropping Database Links

You can drop a database link with the DROP DATABASE LINK statement. For example, to drop the public database link named `DBLINK`, enter the statement:

```
DROP PUBLIC DATABASE LINK dblink;
```

Attention: A database link should not be dropped if it is required to resolve an in-doubt distributed transaction. Refer to the *Oracle Database Administrator's Guide* for additional information about dropping database links.

13.1.4 Examining Available Database Links

The data dictionary of each database stores the definitions of all the database links in that database. Your USER_DB_LINKS data dictionary view shows your defined database links. The ALL_DB_LINKS data dictionary views show all accessible (public and private) database links.

13.1.5 Limiting the Number of Active Database Links

You can limit the number of connections from a user process to remote databases with the parameter OPEN_LINKS. This parameter controls the number of remote connections that any single user process can use concurrently with a single SQL statement. Refer to the *Oracle Database Reference* for additional information about limiting the number of active database links.

13.2 Accessing the Gateway

To access the gateway, complete the following steps on the Oracle integrating server:

13.2.1 Step 1: Login to the Oracle Integrating Server

Login

13.2.2 Step 2: Create a database link to the DRDA database

For example, use:

```
CREATE PUBLIC DATABASE LINK DRDA
CONNECT TO ORADRDA IDENTIFIED BY oracle_pw
USING 'tns_name_entry'
```

13.2.3 Step 3: Retrieve data from the DRDA database

This query fetches the TABLE file in the library SECURE, using the name ORACLE as the DRDA Server user profile. The ORACLE user profile must have the appropriate privilege on the DRDA Server to access the SECURE.TABLE files:

```
SELECT * FROM SECURE.TABLE@DRDA
```

Messages similar to the following are displayed if insufficient privileges were granted to ORACLE:

```
ORA-1031: insufficient privileges
TG4DRDA V10.1.0.2.0 grc=0, drc=-777 (83TC,0000), errp=ARIXO,
sqlcode=-551, sqlstate=42501, errd=FFFFFF9C,0,0,0,0,0
errmc=USER SELECT SECURE.TABLE
```

13.3 Accessing AS/400 File Members

Nothing specific to DRDA or to the gateway allows or disallows access to AS/400 files and file members. However, DB2/400 uses a naming convention that implies that the file member name is the same as the name of the file being addressed. For example, accessing "schema.table" implies that "table" is the file name and also that "table" is the file member name being accessed.

To access file members with names that differ from the associated file name, you must create a view within the file so that DB2/400 can reference the correct file member.

One method for creating this view involves issuing the console command Create Logical File (CRTLF). This action creates a logical association between the file name and the file member name.

For additional information, refer to the AS/400 Command documentation or to the DB2/400 SQL reference document.

13.4 Using the Synonym Feature

You can provide complete data, location, and network transparency by using the synonym feature of the Oracle Database server. When a synonym is defined, the user need not know the underlying table or network protocol being used. A synonym can be public, available to all Oracle users. A synonym can also be defined as private, available only to the user who created it. Refer to the *Oracle Database Reference* for details on the synonym feature.

The following statement creates a system-wide synonym for the EMP file in the DRDA Server with ownership of ORACLE:

```
CREATE PUBLIC SYNONYM EMP FOR ORACLE.EMP@DRDA
```

13.5 Performing Distributed Queries

The Oracle Transparent Gateway technology enables the execution of distributed queries that join Oracle Database servers and DRDA Servers, and any other data store for which Oracle Corporation provides a gateway. These complex operations can be completely transparent to the users requesting the data.

The distributed query optimizer (DQO) capability can provide better performance of distributed queries. Statistical data regarding tables from DRDA Server is retrieved and passed to the Oracle integrating Server. The DQO capability is turned on and off by the DRDA_OPTIMIZE_QUERY parameter. Refer to ["DRDA_OPTIMIZE_QUERY"](#) on page C-7 for more information.

13.5.1 Example of a Distributed Query

The following example joins data between an Oracle Database server, DB2/OS390, and a DRDA Server:

```
SELECT o.custname, p.projno, e.ename, sum(e.rate*p.hours)
FROM orders@DB2 o, EMP@ORACLE7 e, projects@DRDA p
WHERE o.projno = p.projno
AND p.empno = e.empno
GROUP BY o.custname, p.projno, e.ename
```

A combination of views and synonyms, using the following SQL statements, keeps the process of distributed queries transparent to the user:

```
CREATE SYNONYM orders for orders@DB2;
CREATE SYNONYM PROJECTS for PROJECTS@DRDA;
CREATE VIEW details (custname,projno,ename,spend)
AS
SELECT o.custname, p.projno, e.ename, sum(e.rate*p.hours)
FROM orders o, EMP e, projects p
WHERE o.projno = p.projno
AND p.empno = e.empno
GROUP BY o.custname, p.projno, e.ename;
```

This SQL statement retrieves information from these three data stores in one command:

```
SELECT * FROM DETAILS;
```

The results of this command are:

CUSTNAME	PROJNO	ENAME	SPEND
-----	-----	-----	-----
ABC Co.	1	Jones	400
ABC Co.	1	Smith	180
XYZ Inc.	2	Jones	400
XYZ Inc.	2	Smith	180

13.5.2 Two-Phase Commit Processing

To fully participate in a two-phase commit transaction, a server must support the PREPARE TRANSACTION statement. The PREPARE TRANSACTION statement ensures that all participating databases are prepared to COMMIT or to ROLLBACK a specific unit of work.

The Oracle Database server supports the PREPARE TRANSACTION statement. Any number of Oracle Database servers can participate in a distributed two-phase commit transaction. The PREPARE TRANSACTION statement is performed automatically when a COMMIT is issued explicitly by an application or implicitly at the normal end of the application. No other action is needed.

The gateway does not support the PREPARE TRANSACTION statement limiting the two-phase commit protocol when the gateway participates in a distributed transaction. The gateway becomes the commit focal point site of a distributed transaction. Because the gateway is configured as commit/confirm, it is always the commit point site, regardless of the commit point strength setting. The gateway commits the unit of work after verifying that all Oracle Databases in the transaction have successfully committed their work. Because the gateway must coordinate the distributed transaction, only one gateway can participate in an Oracle two-phase commit transaction.

Two-phase commit transactions are recorded in the ORADRDA.Oracle2PC table, which is created during installation. This table is created when the `o2pc.sql` script is run. The owner of this table also owns the package. Refer to ["DRDA Gateway Package Considerations"](#) on page 12-7 for more information.

13.5.3 Distributed DRDA Transactions

Because the ORACLE2PC table is used to record the status of a gateway transaction, the table must reside at the database where the DRDA update takes place. Therefore, all updates that take place over the gateway must be local to the IBM database.

Note: Updates to the ORACLE2PC table cannot be part of an IBM distributed transaction.

For additional information about the two-phase commit process, refer to the *Oracle Database Administrator's Guide*.

13.6 Read-Only Gateway

The read-only option can provide improved performance and security. This improved performance depends on your configuration and parameter selections. A Gateway Initialization Parameter, `DRDA_READ_ONLY`, is used to control whether the gateway is enabled in this mode.

If you enable the read-only feature, then only queries (SELECT statements) are allowed by the gateway. The capabilities that control whether updates are allowed through the gateway are disabled. These capabilities include INSERT, UPDATE, DELETE and Stored-procedure support (pass-through SQL and DB2 stored procedures). Statements attempting to modify records on the DRDA Server are rejected.

Oracle Corporation recommends that you do not routinely switch between settings of the DRDA_READ_ONLY parameter. If you need both update and DRDA_READ_ONLY functionality, then you should create two separate instances of the gateway with different read-only settings.

13.7 Replicating in a Heterogeneous Environment

Oracle Transparent Gateway for DRDA provides a number of options for replicating Oracle and non-Oracle data throughout the enterprise.

13.7.1 Oracle Database 10g Server Triggers

When updates are made to the Oracle Database server, synchronous copies of Oracle and non-Oracle data can be maintained automatically by using Oracle Database 10g server triggers.

13.7.2 Oracle Snapshots

Oracle Transparent Gateway for DRDA can use the Oracle snapshot feature to automatically replicate non-Oracle data into the Oracle Database server. The complete refresh capability of Oracle Snapshot can be used to propagate a complete copy or a subset of the non-Oracle data into the Oracle Database server at user-defined intervals.

13.8 Copying Data from Oracle Database 10g Server to DRDA Server

The COPY command enables you to copy data from an Oracle Database server to a DRDA Server database. The Oracle SQL command INSERT is not supported. If you use the INSERT command:

```
INSERT INTO DRDA_table SELECT * FROM local_table
```

then the following message is displayed:

```
ORA-2025: All tables in the SQL statement must be at the remote database
```

To copy data from your local database to the DRDA Server, use:

```
COPY FROM username/password@connect_identifier -  
INSERT destination_table -  
USING query
```

For example, to select all rows from the local Oracle EMP table, to insert them into the EMP table on the DRDA Server, and to commit the transaction, use:

```
COPY FROM scott/tiger@ORACLE -  
INSERT scott.EMP@DRDA -  
USING SELECT * FROM EMP
```

The SQL*Plus COPY command supports APPEND, CREATE, INSERT, and REPLACE options. However, INSERT is the only option supported when copying to the DRDA Server. For more information about the COPY command, see the *SQL*Plus User's Guide and Reference*.

13.9 Copying Data from DRDA Server to Oracle Database 10g Server

The CREATE TABLE command enables you to copy data from a DRDA Server database to an Oracle Database server. To create a table on your local database and to insert rows from a DRDA Server table, use:

```
CREATE TABLE table_name  
AS query
```

The following example creates the table EMP in your local Oracle Database and inserts the rows from the EMP table on the DRDA Server:

```
CREATE TABLE EMP  
AS SELECT * FROM scott.EMP@DRDA
```

Alternatively, you can use the SQL*Plus COPY command to copy data from a DRDA Server to an Oracle Database server. For more information about the COPY command, refer to the *SQL*Plus User's Guide and Reference*.

13.10 Tracing SQL Statements

SQL statements issued through the gateway can be changed before reaching the DRDA database. These changes are made to make the format acceptable to the gateway or to make Oracle SQL compatible with DRDA Server SQL. The Oracle integrating server and the gateway can change the statements depending upon the situation.

For various reasons, you might need to assess whether the gateway altered the statement correctly or whether the statement could be rewritten to improve performance. SQL tracing is a feature that allows you to see the changes made to a SQL statement by the Oracle integrating server or the gateway.

SQL tracing reduces gateway performance. Use tracing only while testing and debugging your application. Do not enable SQL tracing when the application is running in a production environment. For more information about enabling SQL tracing, refer to the section on ["SQL Tracing and the Gateway"](#) on page 17-6 in [Chapter 17, "Error Messages, Diagnosis, and Reporting"](#).

Developing Applications

The Oracle Transparent Gateway for DRDA allows applications written for the Oracle Database server to access tables in a DRDA database. This access can be virtually transparent by using synonyms or views of the DRDA tables accessed by a database link. However, fundamental SQL, datatype, and semantic differences exist between the Oracle Database server and DRDA databases. Read this chapter to learn about these differences.

This chapter provides information that is specific to this release of the Oracle Transparent Gateway for DRDA, including the following sections:

- [Gateway Appearance to Application Programs](#) on page 14-2
- [Using Oracle Stored Procedures with the Gateway](#) on page 14-3
- [Using DRDA Server Stored Procedures with the Gateway](#) on page 14-4
- [Database Link Behavior](#) on page 14-6
- [Oracle Database Server SQL Construct Processing](#) on page 14-6
- [Native Semantics](#) on page 14-19
- [DRDA Datatype to Oracle Datatype Conversion](#) on page 14-22
- [Passing Native SQL Statements through the Gateway](#) on page 14-28
- [Oracle Data Dictionary Emulation on a DRDA Server](#) on page 14-30
- [Defining the Number of DRDA Cursors](#) on page 14-30

14.1 Gateway Appearance to Application Programs

An application that is written to access information in a DRDA database interfaces with an Oracle integrating server. When developing applications, keep the following information in mind:

- You must define the DRDA database to the application by the use of a database link that is defined at the Oracle integrating server. Your application specifies tables that exist on a DRDA database by using the name that is defined in the database link. For example, assume that a database link is defined such that it names the DRDA database link `DRDA`, and also assume that an application needs to retrieve data from an Oracle Database and from the `DRDA` database. Use the following SQL statement (joining two tables together) in your application:

```
SELECT EMPNO, SALARY
FROM EMP L, EMPS@DRDA R
WHERE L.EMPNO = R.EMPNO
```

In this example, `EMP` is a table on an Oracle Database server, and `EMPS` is a table on a DRDA Server. You can also define a synonym or a view on the DRDA Server table, and access the information without the database link suffix.

- You can perform reads and writes of data to a defined DRDA database. `SELECT`, `INSERT`, `UPDATE`, and `DELETE` are all valid operations.
- A single transaction can write to one DRDA database and to multiple Oracle Databases.
- Single SQL statements, using `JOINS`, can refer to tables in multiple Oracle Databases or in multiple DRDA databases, or in both.

14.1.1 Fetch Reblocking

The Oracle Database server supports fetch reblocking with the `HS_RPC_FETCH_REBLOCKING` parameter.

When the value of this parameter is set to `ON` (the default), the array size for `SELECT` statements is determined by the `HS_RPC_FETCH_SIZE` value. The `HS_RPC_FETCH_SIZE` parameter defines the number of bytes sent with each buffer from the gateway to the Oracle Database 10g server. The buffer might contain one or more qualified rows from the DRDA Server. This feature can provide significant performance enhancements, depending upon your application design, installation type, and workload.

The array size between the client and the Oracle Database 10g server is still determined by the Oracle application.

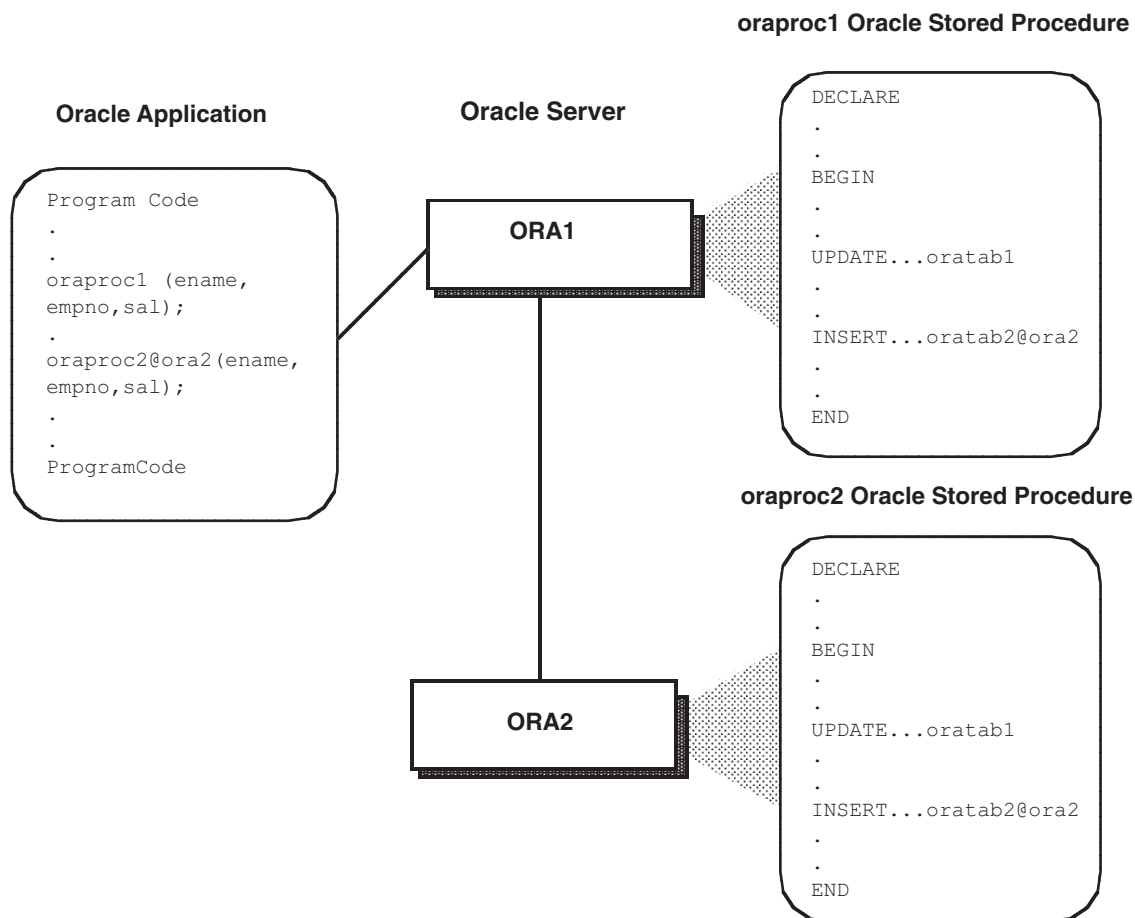
Refer to [Chapter 12, "Configuring the Gateway"](#), for more information.

14.2 Using Oracle Stored Procedures with the Gateway

The gateway stored procedure support is an extension of Oracle stored procedures. An Oracle stored procedure is a schema object that logically groups together a set of SQL and other PL/SQL programming language statements to perform a specific task. Oracle stored procedures are stored in the database for continued use. Applications use standard Oracle PL/SQL to call stored procedures.

Oracle stored procedures can be located in a local instance of the Oracle Database server and in a remote instance. [Figure 14–1, "Calling Oracle Stored Procedures in a Distributed Oracle Environment"](#) illustrates two stored procedures: `oraproc1` is a procedure stored in the ORA1 Oracle instance, while `oraproc2` is a procedure stored in the ORA2 Oracle instance.

Figure 14–1 Calling Oracle Stored Procedures in a Distributed Oracle Environment



To maintain location transparency in the application, a synonym can be created:

```
CREATE SYNONYM oraproc2 FOR oraproc2@ora2;
```

After this synonym is created, the application no longer needs to use the database link specification to call the stored procedure at the remote Oracle instance.

In [Figure 14–1](#), the second statement in `oraproc1` is used to access a table in the ORA2 instance. In the same way, Oracle stored procedures can be used to access DB2 tables through the gateway.

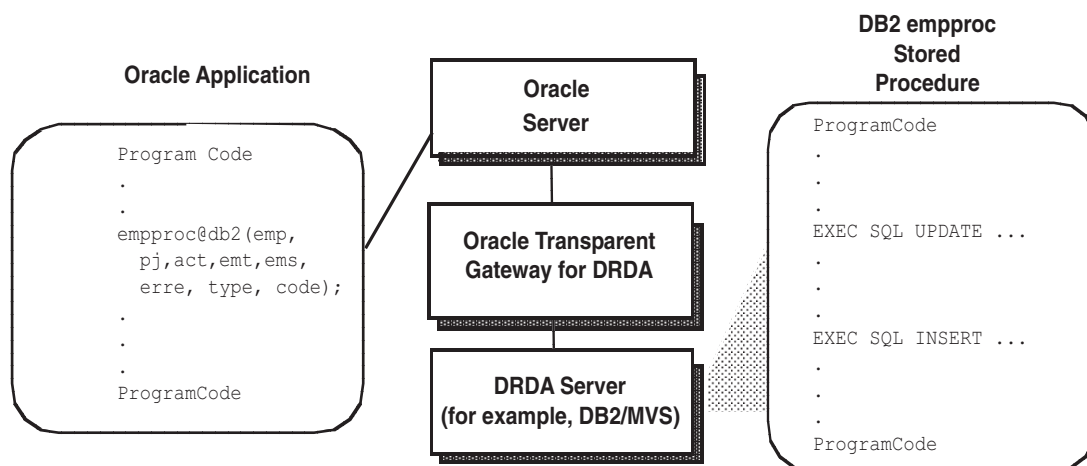
14.3 Using DRDA Server Stored Procedures with the Gateway

The procedural feature of the gateway enables invocation of native DRDA Server stored procedures. In other words, the stored procedure is no longer defined in the Oracle Database server, but instead, is defined to the DRDA Server (for example., DB2/OS390). Again, standard Oracle PL/SQL is used by the Oracle application to run the stored procedure.

After the stored procedure is defined to the DRDA Server (for example., DB2/OS390), the gateway is able to use the existing DRDA Server definition to run the procedure. The gateway does not require special definitions to call the DB2 stored procedure.

In [Figure 14-2](#), an Oracle application calls the `empproc` stored procedure that is defined to the DRDA Server (for example., DB2/OS390).

Figure 14-2 Running DRDA Server Stored Procedures



From the perspective of the application, running the DB2 stored procedure is no different from invoking a stored procedure at a remote Oracle Database server instance.

14.3.1 Oracle Application and DRDA Server Stored Procedure Completion

As an example, suppose an Oracle Application attempts to invoke a stored procedure in a DB2/OS390 database. In order for an Oracle application to call a DB2 stored procedure, it is first necessary to create the DB2 stored procedure on the DB2 system by using the procedures documented in the IBM reference document for DB2 for OS/390 SQL.

After the stored procedure is defined to DB2, the gateway is able to access the data using a standard PL/SQL call. For example, an employee name, JOHN SMYTHE, is passed to the DB2 stored procedure REVISE_SALARY. The DB2 stored procedure retrieves the salary value from the DB2 database in order to calculate a new yearly salary for JOHN SMYTHE. The revised salary that is returned in RESULT is used to update the EMP table of an Oracle Database server:

```
DECLARE
  INPUT VARCHAR2(15);
  RESULT NUMBER(8,2);
BEGIN
  INPUT := 'JOHN SMYTHE';
  REVISE_SALARY@DB2(INPUT, RESULT);
  UPDATE EMP SET SAL = RESULT WHERE ENAME = INPUT;
END;
```

When the gateway receives a call to run a stored procedure on the DRDA Server (for example, DB2/OS390), it first does a lookup of the procedure name in the server catalog. The information that defines a stored procedure is stored in different forms on each DRDA Server. For example, DB2/OS390 V5.0 uses the table SYSIBM.SYSPROCEDURES, while DB2/OS390 V6.1 uses the table SYSIBM.SYSROUTINES and SYSIBM.SYSPARMS, and DB2/400 uses the table QSYS2.SYSPROCS and QSYS2.SYSPARMS. The gateway has a list of known catalogs to search, depending upon the DRDA Server that is being accessed.

The search order of the catalogs is dependent on whether the catalogs support Location designators (such as LUNAME in SYSIBM.SYSPROCEDURES), and Authorization or Owner IDs (such as AUTHID in SYSIBM.SYSPROCEDURES or OWNER in SYSIBM.SYSROUTINES).

Some DRDA Servers allow blank or public Authorization qualifiers. If the DRDA Server currently that is connected supports this form of qualification, the gateway will apply those naming rules when searching for a procedure name in the catalog.

The matching rules will first search for a Public definition, and then an Owner qualified procedure name. For more detailed information, refer to the IBM reference document for DB2 for OS/390 SQL.

14.3.2 Procedural Feature Considerations with DB2

The following are special considerations for using the procedural feature with the gateway:

- DB2 stored procedures do not have the ability to coordinate, commit, and rollback activity on recoverable resources such as IMS or CICS transactions. Therefore, if the DB2 stored procedure calls a CICS or IMS transaction, then it is considered a separate unit of work and does not affect the completion of the stored procedure. This means that if you are running a DB2 stored procedure from an Oracle application, and if this procedure calls a CICS or IMS transaction, then the gateway cannot recover from any activity that occurred within the CICS or IMS transaction.

For example, the CICS transaction could roll back a unit of work, but this does not prevent the gateway from committing other DB2 work contained within the DB2 stored procedure.

Likewise, if the DB2 stored procedure updated an irrecoverable resource such as a VSAM file, then the gateway would consider this activity separate from its own recoverable unit of work.

- PL/SQL records cannot be passed as parameters when invoking a DB2 stored procedure.
- The gateway supports the SIMPLE linkage convention of DB2 stored procedures. The SIMPLE linkage convention means that the parameters that are passed to and from the DB2 stored procedure cannot be null.

14.4 Database Link Behavior

A connection to the gateway is established through a database link when it is first used in an Oracle session. In this context, connection refers to both the connection between the Oracle integrating server and the gateway, and to the DRDA network connection between the gateway and the target DRDA database. The connection remains established until the Oracle session ends. Another session or user can access the same database link and get a distinct connection to the gateway and DRDA database.

Connections to the DRDA database can be limited in an APPC configuration in a parallel session limit, or by other factors, such as memory, gateway parameters, or DRDA Server resources. In a TCP/IP configuration, only resource limits (such as memory) or limits on the number of connections by the DRDA Server will limit the number of connections between the gateway and the DRDA Server.

14.5 Oracle Database Server SQL Construct Processing

One of the most important features of the Oracle Open Gateways family of products is providing SQL transparency to the user and to the application programmer. Foreign SQL constructs can be categorized into four areas:

- Compatible
- Translated
- Compensated
- Native semantics

14.5.1 Compatible SQL Functions

The Oracle integrating server automatically forwards to the DRDA database compatible SQL functions—that is, SQL constructs with the same syntax and meaning on both the Oracle Database server and the DRDA database. These SQL constructs are forwarded unmodified. All of the compatible functions are column functions. Functions that are not compatible are either translated to an equivalent DRDA SQL function or are compensated (post-processed) by the Oracle Database server after the data is returned from the DRDA database.

14.5.2 Translated SQL Functions

Translated functions have the same meaning but different names between the Oracle integrating server and the DRDA database, but all applications must use the Oracle function name. These SQL constructs that are supported with different syntax (such as different function names) by the DRDA database, are automatically translated by the Oracle Database server and then forwarded to the DRDA database. The Oracle integrating server (in a manner that is transparent to your application) changes the function name before sending it to the DRDA database.

14.5.3 Compensated SQL Functions

Some advanced SQL constructs that are supported by the Oracle Database server may not be supported in the same manner, if at all, by the DRDA database. Compensated functions are those SQL functions that are not recognized by the DRDA Server. If a SELECT statement containing one of these functions is passed from the Oracle integrating server to the gateway, then the gateway removes the function before passing the SQL statement to the DRDA Server. The gateway passes the selected DRDA database rows to the Oracle integrating server. The Oracle integrating server then applies the function.

14.5.3.1 Post-Processing

The Oracle Database server can compensate for a missing or incompatible function by automatically excluding the incompatible SQL construct from the SQL request that is forwarded to the DRDA database. The Oracle Database server then retrieves the necessary data from the DRDA database and applies the function. This process is known as post-processing.

The gateway attempts to pass all SQL functions to DRDA databases. But when a DRDA database does not support a function that is represented in the computation, then the gateway changes that function. For example, if a program requests:

```
SELECT COS(X_COOR) FROM TABLE_X;
```

from a DB2/OS390 database, which does not support the meaning of COS, then the gateway changes the SELECT statement to:

```
SELECT X_COOR FROM TABLE_X;
```

All data in the X_COOR column of TABLE_X is passed from the DB2/OS390 database to the Oracle integrating server. After the data is moved to the Oracle integrating server, the COS function is performed.

If you are performing operations on large amounts of data that are stored in a DRDA database, then keep in mind that some functions require post-processing.

14.5.4 Native Semantic SQL Functions

Some SQL functions that are normally compensated may also be overridden, via the Native Semantics facility. If a SQL function has been enabled for Native Semantics, then the function may be passed on to the DRDA database for processing, instead of being compensated (post-processed). If a SQL function is enabled for Native Semantics and is therefore passed on the DRDA database for processing, then we describe the operation by saying that the SQL function was processed **natively** in the DRDA database. Refer to ["Native Semantics"](#) on page 14-19 for more information.

14.5.5 DB2/OS390 SQL Compatibility

The ways that the Oracle Database server and gateway handle SQL functions for a DB2/OS390 database are shown in the following table:

Table 14–1 DB2/OS390 SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
ABS			Yes	Yes
ACOS			Yes	Yes
ADD_MONTHS			Yes	
ASCII			Yes	Yes
ASIN			Yes	Yes
ATAN			Yes	Yes
ATAN2			Yes	Yes
AVG	Yes			
CAST			Yes	Yes
CEIL			Yes	Yes
CHARTOROWID			Yes	
CHR			Yes	Yes
CONCAT	Yes			
CONVERT			Yes	Yes
COS			Yes	Yes
COSH			Yes	Yes
COUNT(*)	Yes			
DECODE			Yes	Yes
DUMP			Yes	Yes
EXP			Yes	Yes
FLOOR			Yes	Yes
GREATEST			Yes	Yes
HEXTORAW			Yes	Yes
INITCAP			Yes	Yes

Table 14–1 (Cont.) DB2/OS390 SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
INSTR			Yes	Yes
INSTRB			Yes	Yes
LAST_DAY			Yes	
LEAST			Yes	Yes
LENGTH			Yes	Yes
LENGTHB			Yes	Yes
LN			Yes	Yes
LOG			Yes	Yes
LOWER			Yes	Yes
LPAD			Yes	Yes
LTRIM			Yes	Yes
MAX	Yes			
MIN	Yes			
MOD			Yes	Yes
MONTHS_BETWEEN			Yes	
NEW_TIME			Yes	
NEXT_DAY			Yes	
NLS_INITCAP			Yes	Yes
NLS_LOWER			Yes	Yes
NLS_UPPER			Yes	Yes
NLSSORT			Yes	Yes
NVL		VALUE		
NVL2			Yes	Yes
POWER			Yes	Yes
RAWTOHEX			Yes	Yes
REPLACE			Yes	Yes
REVERSE			Yes	Yes
ROUND			Yes	Yes
ROWIDTOCHAR			Yes	
RPAD			Yes	Yes
RTRIM			Yes	Yes
SIGN			Yes	Yes
SIN			Yes	Yes
SINH			Yes	Yes

Table 14–1 (Cont.) DB2/OS390 SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
SOUNDEX			Yes	
SQRT			Yes	Yes
STDDEV			Yes	Yes
SUBSTR			Yes	Yes
SUBSTRB			Yes	Yes
SUM	Yes			
SYSDATE			Yes	
TAN			Yes	Yes
TANH			Yes	Yes
TO_CHAR			Yes	
TO_DATE			Yes	
TO_LABEL			Yes	
TO_MULTI_BYTE			Yes	
TO_NUMBER		DECIMAL		Yes
TO_SINGLE_BYTE			Yes	
TRANSLATE			Yes	Yes
TRIM		STRIP	Yes	Yes
TRUNC			Yes	Yes
UID			Yes	
UPPER			Yes	
USER			Yes	
USERENV			Yes	
VARIANCE			Yes	Yes
VSIZE			Yes	Yes

14.5.6 DB2/Universal Database SQL Compatibility

The ways that the Oracle Database server and gateway handle SQL functions for a DB2/UDB database are shown in the following table:

Table 14–2 DB2/Universal Database SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
ABS	Yes			Yes
ACOS			Yes	Yes
ADD_MONTHS			Yes	

Table 14–2 (Cont.) DB2/Universal Database SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
ASCII			Yes	Yes
ASIN			Yes	Yes
ATAN			Yes	Yes
ATAN2			Yes	Yes
AVG	Yes			
CAST			Yes	Yes
CEIL	Yes			Yes
CHARTOROWID			Yes	
CHR	Yes			Yes
CONCAT	Yes			
CONVERT			Yes	Yes
COS	Yes			Yes
COSH			Yes	Yes
COUNT(*)	Yes			
DECODE			Yes	Yes
DUMP			Yes	Yes
EXP	Yes			Yes
FLOOR	Yes			Yes
GREATEST			Yes	Yes
HEXTORAW			Yes	Yes
INITCAP			Yes	Yes
INSTR			Yes	Yes
INSTRB			Yes	Yes
LAST_DAY			Yes	
LEAST			Yes	Yes
LENGTH			Yes	Yes
LENGTHB			Yes	Yes
LN	Yes			Yes
LOG			Yes	Yes
LOWER		LCASE		Yes
LPAD			Yes	Yes
LTRIM			Yes	Yes
MAX	Yes			
MIN	Yes			

Table 14–2 (Cont.) DB2/Universal Database SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
MOD	Yes			Yes
MONTHS_BETWEEN			Yes	
NEW_TIME			Yes	
NEXT_DAY	Yes		Yes	
NLS_INITCAP			Yes	Yes
NLS_LOWER			Yes	Yes
NLS_UPPER			Yes	Yes
NLSSORT			Yes	Yes
NVL		VALUE		
NVL2			Yes	Yes
POWER	Yes			Yes
RAWTOHEX			Yes	Yes
REPLACE	Yes			Yes
REVERSE			Yes	Yes
ROUND	Yes			Yes
ROWIDTOCHAR			Yes	
RPAD			Yes	Yes
RTRIM			Yes	Yes
SIGN	Yes			Yes
SIN	Yes			Yes
SINH			Yes	Yes
SOUNDEX			Yes	
SQRT	Yes			Yes
STDDEV			Yes	Yes
SUBSTR			Yes	Yes
SUBSTRB			Yes	Yes
SUM	Yes			
SYSDATE			Yes	
TAN	Yes			Yes
TANH			Yes	Yes
TO_CHAR			Yes	
TO_DATE			Yes	
TO_LABEL			Yes	
TO_MULTI_BYTE		Yes	Yes	

Table 14–2 (Cont.) DB2/Universal Database SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
TO_NUMBER		DECIMAL		Yes
TO_SINGLE_BYTE			Yes	
TRANSLATE			Yes	Yes
TRIM			Yes	Yes
TRUNC	Yes			Yes
UID			Yes	
UPPER		UCASE		Yes
USER			Yes	
USERENV			Yes	
VARIANCE			Yes	Yes
VSIZE			Yes	Yes

14.5.7 DB2/400 SQL Compatibility

The ways that the Oracle Database server and gateway handle SQL functions for a DB2/400 database are shown in the following table:

Table 14–3 DB2/400 SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
ABS		ABSVAL		Yes
ACOS			Yes	Yes
ADD_MONTHS			Yes	
ASCII			Yes	Yes
ASIN			Yes	Yes
ATAN			Yes	Yes
ATAN2			Yes	Yes
AVG	Yes			
CAST			Yes	Yes
CEIL			Yes	Yes
CHARTOROWID			Yes	
CHR			Yes	Yes
CONCAT	Yes			
CONVERT			Yes	Yes
COS	Yes			Yes
COSH	Yes			Yes

Table 14–3 (Cont.) DB2/400 SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
COUNT(*)	Yes			
DECODE			Yes	Yes
DUMP			Yes	Yes
EXP	Yes			Yes
FLOOR			Yes	Yes
GREATEST			Yes	Yes
HEXTORAW			Yes	Yes
INITCAP			Yes	Yes
INSTR			Yes	Yes
INSTRB			Yes	Yes
LAST_DAY			Yes	
LEAST			Yes	Yes
LENGTH			Yes	Yes
LENGTHB			Yes	Yes
LN	Yes			Yes
LOG			Yes	Yes
LOWER			Yes	Yes
LPAD			Yes	Yes
LTRIM			Yes	Yes
MAX	Yes			
MIN	Yes			
MOD			Yes	Yes
MONTHS_BETWEEN			Yes	
NEW_TIME			Yes	
NEXT_DAX			Yes	
NLS_INITCAP			Yes	Yes
NLS_LOWER			Yes	Yes
NLS_UPPER			Yes	Yes
NLSSORT			Yes	Yes
NVL		VALUE		
NVL2			Yes	Yes
POWER			Yes	Yes
RAWTOHEX			Yes	Yes
REPLACE			Yes	Yes

Table 14–3 (Cont.) DB2/400 SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
REVERSE			Yes	Yes
ROUND			Yes	Yes
ROWIDTOCHAR			Yes	
RPAD			Yes	Yes
RTRIM			Yes	Yes
SIGN			Yes	Yes
SIN	Yes			Yes
SINH	Yes			Yes
SOUNDEX			Yes	
SQRT	Yes			Yes
STDDEV	Yes			Yes
SUBSTR			Yes	Yes
SUBSTRB			Yes	Yes
SUM	Yes			
SYSDATE			Yes	
TAN	Yes			Yes
TANH	Yes			Yes
TO_CHAR			Yes	
TO_DATE			Yes	
TO_LABEL			Yes	
TO_MULTI_BYTE			Yes	
TO_NUMBER			Yes	Yes
TO_SINGLE_BYTE			Yes	
TRANSLATE			Yes	Yes
TRIM			Yes	Yes
TRUNC			Yes	Yes
UID			Yes	
UPPER		TRANSLATE		Yes
USER			Yes	
USERENV			Yes	
VARIANCE		VAR		Yes
VSIZE			Yes	Yes

14.5.8 DB2/VM SQL Compatibility

The ways that the Oracle Database server and gateway handle SQL functions for a DB2/VM database are shown in the following table:

Table 14–4 DB2/VM SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
ABS			Yes	Yes
ACOS			Yes	Yes
ADD_MONTHS			Yes	
ASCII			Yes	Yes
ASIN			Yes	Yes
ATAN			Yes	Yes
ATAN2			Yes	Yes
AVG	Yes			
CAST			Yes	Yes
CEIL			Yes	Yes
CHARTOROWID			Yes	
CHR			Yes	Yes
CONCAT	Yes			
CONVERT			Yes	Yes
COS			Yes	Yes
COSH			Yes	Yes
COUNT(*)	Yes			
DECODE			Yes	Yes
DUMP			Yes	Yes
EXP			Yes	Yes
FLOOR			Yes	Yes
GREATEST			Yes	Yes
HEXTORAW			Yes	Yes
INITCAP			Yes	Yes
INSTR			Yes	Yes
INSTRB			Yes	Yes
LAST_DAY			Yes	
LEAST			Yes	Yes
LENGTH			Yes	Yes
LENGTHB			Yes	Yes
LN			Yes	Yes

Table 14–4 (Cont.) DB2/VM SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
LOG			Yes	Yes
LOWER			Yes	Yes
LPAD			Yes	Yes
LTRIM			Yes	Yes
MAX	Yes			
MIN	Yes			
MOD			Yes	Yes
MONTHS_BETWEEN			Yes	
NEW_TIME			Yes	
NEXT_DAY			Yes	
NLS_INITCAP			Yes	Yes
NLS_LOWER			Yes	Yes
NLS_UPPER			Yes	Yes
NLSSORT			Yes	Yes
NVL		VALUE		
NVL2			Yes	Yes
POWER			Yes	Yes
RAWTOHEX			Yes	Yes
REPLACE			Yes	Yes
REVERSE			Yes	Yes
ROUND			Yes	Yes
ROWIDTOCHAR			Yes	
RPAD			Yes	Yes
RTRIM			Yes	Yes
SIGN			Yes	Yes
SIN			Yes	Yes
SINH			Yes	Yes
SOUNDEX			Yes	
SQRT			Yes	Yes
STDDEV			Yes	Yes
SUBSTR			Yes	Yes
SUBSTRB			Yes	Yes
SUM	Yes			
SYSDATE			Yes	

Table 14–4 (Cont.) DB2/VM SQL Compatibility, by Oracle SQL Function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
TAN			Yes	Yes
TANH			Yes	Yes
TO_CHAR			Yes	
TO_DATE			Yes	
TO_LABEL			Yes	
TO_MULTI_BYTE			Yes	
TO_NUMBER			Yes	Yes
TO_SINGLE_BYTE			Yes	
TRANSLATE			Yes	Yes
TRIM			Yes	Yes
TRUNC			Yes	Yes
UID			Yes	
UPPER			Yes	Yes
USER			Yes	
USERENV			Yes	
VARIANCE			Yes	Yes
VSIZE			Yes	Yes

14.6 Native Semantics

Because some of the advanced SQL constructs that are supported by the Oracle Database server may not be supported in the same manner (if at all) by the DRDA database, the Oracle Database server compensates for the missing or incompatible functionality by post-processing the DRDA database data with Oracle Database server functionality (Refer to the previous section, "[Oracle Database Server SQL Construct Processing](#)" on page 14-6 for more information). This feature provides maximum transparency, but may impact performance. In addition, new versions of a particular DRDA database may implement previously unsupported functions or capabilities, or they may change the supported semantics in such a manner as to make them more compatible with Oracle Database server functions.

Some of the DRDA Servers also provide support for user-defined functions. The user may choose to implement Oracle Database server functions natively (in other words, in the DRDA database), thus allowing the DRDA Server to pass the function on to the underlying database implementation (for example, DB2). Native Semantics provides a method of allowing specific capabilities to be processed natively by the DRDA Server.

Various considerations must be taken into account when enabling the Native Semantic feature of a particular function because Native Semantics has advantages and disadvantages, which are typically a trade-off between transparency and performance. One such consideration is transparency of data coercion. The Oracle Database server provides coercion (implicit data conversion) for many SQL functions. This means that if the supplied value for a particular function is not correct, then Oracle will coerce the value (change it to the correct value type) before processing it. However, with the Native Semantic feature enabled, the value (exactly as provided) will be passed through to the DRDA Server for processing. In many cases, the DRDA Server will not be able to coerce the value to the correct type and will generate an error.

Another consideration involves the compatibility of parameters to a particular SQL function. For instance, the Oracle Database server implementation of SUBSTR allows negative values for the string index, whereas most DRDA Server implementations of SUBSTR do not allow negative values for the string index. However, if the application is implemented to invoke SUBSTR in a manner that is compatible with the DRDA Server, then the function will behave the same in either the Oracle Database server or the DRDA Server.

Another consideration is that the processing of a function at the DRDA Server may not be desirable due to resource constraints in that environment.

Refer to the "[DRDA_CACHE_TABLE_DESC](#)" parameter on page C-2 for details on enabling or disabling these capabilities. Refer to the *Oracle Database SQL Reference* for the Oracle Database server format of the following capabilities.

14.6.1 SQL Functions That Can Be Enabled

The following list contains SQL functions that are disabled (**OFF**) by default. They can be enabled (turned **ON**) as an option:

- ABS
- ACOS
- ASCII
- ASIN
- ATAN
- ATAN2

- CAST
- CEIL
- CHR
- CONVERT
- COS
- COSH
- DECODE
- DUMP
- EXP
- FLOOR
- GREATEST
- HEXTORAW
- INITCAP
- INSTR
- INSTRB
- LEAST
- LENGTH
- LENGTHB
- LN
- LOG
- LOWER
- LPAD
- LTRIM
- MOD
- NLS_INITCAP
- NLS_UPPER
- NLS_LOWER
- NLSSORT
- NVL2
- POWER
- RAWTOHEX
- REPLACE
- REVERSE
- ROUND
- RPAD
- RTRIM
- SIGN

- SIN
- SINH
- SQRT
- STDDEV
- SUBSTR
- SUBSTRB
- TAN
- TANH
- TO_NUMBER
- TRANSLATE
- TRIM
- TRUNC
- UPPER
- VARIANCE
- VSIZE

14.6.2 SQL Functions That Can Be Disabled

The following list shows the SQL functions that are enabled (ON) by default. They can be disabled (turned OFF) as an option:

- GROUPBY
- HAVING
- ORDERBY
- WHERE

ORDERBY controls sort order, which may differ at various sort locations. For example, with ORDERBY ON, a DB2 sort would be based on EBCDIC sorting order, whereas with ORDERBY OFF, an Oracle sort would be based on ASCII sorting order.

The other three functions, GROUPBY, HAVING, and WHERE, can take additional processing time. If you need to minimize the use of expensive resources, you should choose the settings of these functions so that the processing is performed on the cheaper resource.

14.6.3 SQL Set Operators and Clauses

The clauses WHERE and HAVING are compatible for all versions of the DRDA Server, meaning that they are passed unchanged to the DRDA Server for processing. Whether clauses GROUP BY and ORDER BY are passed to the DRDA Server, or compensated by the Oracle Database server, is determined by the Native Semantics Parameters (see the previous section).

The set operators UNION and UNION ALL are compatible for all versions of the DRDA Server, meaning that they are passed unchanged to the DRDA Server for processing. The set operators INTERSECT and MINUS are compensated on all versions of the DRDA Server except DB2/UDB. For DB2/UDB, INTERSECT is compatible and MINUS is translated to EXCEPT.

14.7 DRDA Datatype to Oracle Datatype Conversion

To move data between applications and the database, the gateway binds data values from a host variable or literal of a specific datatype to a datatype understood by the database. Therefore, the gateway maps values from any version of the DRDA Server into appropriate Oracle datatypes before passing these values back to the application or Oracle tool.

The following table lists the datatype mapping and restrictions. The DRDA Server datatypes that are listed in the table are general. Refer to documentation for your DRDA database for restrictions on datatype size and value limitations.

Table 14–5 Datatype Mapping and Restrictions

DRDA Server	Oracle External	Criteria
CHAR(N)	CHAR(N)	N <= 255
VARCHAR (N)	VARCHAR2(N) LONG	N <= 2000 2000 < N <= 32740
LONG VARCHAR(N)	VARCHAR2(N)	N <= 2000
LONG VARCHAR(N)	LONG	2000 < N <= 32740
CHAR(N) FOR BIT DATA	RAW(N)	N <= 255
VARCHAR(N) FOR BIT DATA	RAW(N)	1 <= N <= 255
VARCHAR(N) FOR BIT DATA	LONG RAW(N)	255 < N <= 32740
LONG VARCHAR(N) FOR BIT DATA	RAW(N)	1 <= N <= 255
LONG VARCHAR(N) FOR BIT DATA	LONG RAW(N)	255 < N <= 32740
DATE	DATE	Refer to Performing Date and Time Operations
TIME	CHAR(8)	See Performing Date and Time Operations
TIMESTAMP	CHAR(26)	See Performing Date and Time Operations
GRAPHIC	CHAR(2N)	N <= 127
VARGRAPHIC	VARCHAR2(2N) LONG	N <= 1000 1000 <= N <= 16370
LONG VARGRAPHIC	VARCHAR2(2N) LONG	N <= 1000 1000 <= N <= 16370
Floating Point Single	FLOAT(21)	n/a
Floating Point Double	FLOAT(53)	n/a
Decimal (P, S)	NUMBER(P,S)	n/a

14.7.1 Performing Character String Operations

The gateway performs all character string comparisons, concatenations, and sorts using the datatype of the referenced columns, and determines the validity of character string values passed by applications using the gateway. The gateway automatically converts character strings from one datatype to another and converts between character strings and dates when needed.

Frequently, DRDA databases are designed to hold non-character binary data in character columns. Applications executed on DRDA systems can generally store and retrieve data as though it contained character data. However, when an application accessing this data runs in an environment that uses a different character set, inaccurate data might be returned.

With the gateway running on the host, character data retrieved from a DB2/400, DB2/OS390, or DB2/VM host is translated from EBCDIC to ASCII. When character data is sent to DB2/400, DB2/OS390, or DB2/VM from the host, ASCII data is translated to EBCDIC. When the characters are binary data in a character column, this translation causes the application to receive incorrect information or errors. To resolve these errors, character columns on DB2/400, DB2/OS390, or DB2/VM that hold non-character data must be created with the FOR BIT DATA option. In the application, the character columns holding non-character data should be processed using the Oracle datatypes RAW and LONG RAW. The DESCRIBE information for a character column defined with FOR BIT DATA on the host always indicates RAW or LONG RAW.

14.7.2 Converting Character String Datatypes

The gateway binds character string data values from host variables as fixed-length character strings. The bind length is the length of the character string data value. The gateway performs this conversion on every bind.

The DRDA VARCHAR datatype can be from 1 to 32740 bytes in length. This datatype is converted to an Oracle VARCHAR2 datatype if it is between 1 and 2000 characters in length. If it is between 2000 and 32740 characters in length, it is converted to an Oracle LONG datatype.

The DRDA VARCHAR datatype can be no longer than 32740 bytes, which is much shorter than the maximum size for the Oracle LONG datatype. If you define an Oracle LONG datatype larger than 32740 bytes in length, you receive an error message when it is mapped to the DRDA VARCHAR datatype.

14.7.3 Performing Graphic String Operations

DB2 GRAPHIC datatypes store only double-byte string data. Sizes for DB2 GRAPHIC datatypes typically have maximum sizes which are half that of their Character counterparts. For example, the maximum size of a CHAR may be 255 characters, whereas the maximum size of a GRAPHIC may be 127 characters.

Oracle does not have a direct matching datatype, and the gateway therefore converts between Oracle Character datatypes and DB2 Graphic datatypes. Oracle character datatypes may contain single, mixed, or double-byte character data. The gateway converts the string data into appropriate double-byte-only format depending upon whether the target DB2 column is a Graphic type and whether gateway initialization parameters are set to perform this conversion. For more configuration information, refer to [Appendix C, "DRDA-Specific Parameters"](#) and [Appendix D, "National Language Support"](#).

14.7.4 Performing Date and Time Operations

The implementation of date and time data differs significantly in IBM DRDA databases and the Oracle Database server. The Oracle Database server has a single date datatype, **DATE**, that can contain both calendar date and time of day information.

IBM DRDA databases support the following three distinct date and time datatypes:

DATE is the calendar date only.

TIME is the time of day only.

TIMESTAMP is a numerical value combining calendar date and time of day with microsecond resolution in the internal format of the IBM DRDA database.

14.7.4.1 Processing TIME and TIMESTAMP Data

There is no built-in mechanism that translates the IBM **TIME** and **TIMESTAMP** data to Oracle **DATE** data. An application must process **TIME** datatypes to the Oracle **CHAR** format with a length of eight bytes. An application must process the **TIMESTAMP** datatype in the Oracle **CHAR** format with a length of 26 bytes.

An application reads **TIME** and **TIMESTAMP** functions as character strings and converts or subsets portions of the string to perform numerical operations. **TIME** and **TIMESTAMP** values can be sent to an IBM DRDA database as character literals or bind variables of the appropriate length and format.

14.7.4.2 Processing DATE Data

Oracle and IBM **DATE** datatypes are mapped to each other. If an IBM **DATE** is queried, then it is converted to an Oracle **DATE** with a zero (midnight) time of day. If an Oracle **DATE** is processed against an IBM **DATE** column, then the date value is converted to the IBM **DATE** format, and any time value is discarded.

Character representations of dates are different in Oracle format and IBM DRDA format. When an Oracle application SQL statement contains a date literal, or conveys a date using a character bind variable, the gateway must convert the date to an IBM DRDA compatible format.

The gateway does not automatically recognize when a character value is going to be processed against an IBM **DATE** column. Applications are required to distinguish character date values by enclosing them with Oracle **TO_DATE** function notation. For example, if **EMP** is a synonym or view that accesses data on an IBM DRDA database, then instead of this SQL statement:

```
SELECT * FROM EMP WHERE HIREDATE = '03-MAR-81'
```

you must use:

```
SELECT * FROM EMP WHERE HIREDATE = TO_DATE('03-MAR-81')
```

In a programmatic interface program that uses a character bind variable for the qualifying date value, you must use this SQL statement:

```
SELECT * FROM EMP WHERE HIREDATE = TO_DATE(:1)
```

The above SQL notation does not affect SQL statement semantics when the statement is executed against an Oracle table. The statement remains portable across Oracle and IBM DRDA-accessed data stores.

The TO_DATE function is not required for dates in any of the following formats:

- YYYY-MM-DD (ISO/JIS)
- DD.MM.YYYY (European)
- MM/DD/YYYY (USA)

For example:

```
SELECT * FROM EMP WHERE HIREDATE = '1981-03-03'
```

The TO_DATE requirement also does not pertain to input bind variables that are in Oracle date 7-byte binary format. The gateway recognizes such values as dates.

14.7.4.3 Performing Date Arithmetic

The following forms of SQL expression generally do not work correctly with the gateway:

```
date + number
number + date
date - number
date1 - date2
```

The date and number addition and subtraction (*date + number*, *number + date*, *date - number*) forms are sent through to the DRDA Server, where they are rejected. The supported servers do not allow number addition or subtraction with dates.

Because of differing interpretations of date subtraction in the supported servers, subtracting two dates (*date1 - date2*) gives results that vary by server.

Note: Avoid date arithmetic expressions in all gateway SQL until date arithmetic problems are resolved.

14.7.5 Dates

Date handling has two categories: two-digit year dates, which are treated as occurring 50 years before or 50 years after the year 2000, and four-digit year dates, which are not ambiguous with regard to the year 2000. Oracle Corporation recommends that you set the Oracle Database 10g server and gateway default HS_NLS_DATE_FORMAT parameter to a format including a four-digit year.

Use one of the following methods to enter twenty-first century dates:

- The TO_DATE function

Use any date format including a four character year field. Refer to the *Oracle Database SQL Reference* for the available date format string options.

For example, TO_DATE('2008-07-23', 'YYYY-MM-DD') can be used in any SELECT, INSERT, UPDATE, or DELETE statement.

- The HS_NLS_DATE_FORMAT parameter

The HS_NLS_DATE_FORMAT parameter defines a default format for the Oracle Database server explicit TO_DATE functions without a pattern and for implicit string to date conversions.

For example, with HS_NLS_DATE_FORMAT defined as 'YYYY-MM-DD', '2008-07-23' can be used in any SELECT, INSERT, UPDATE, or DELETE statement.

14.7.6 HS_NLS_DATE_FORMAT Support

The following table lists the four patterns that can be used for the HS_NLS_DATE_FORMAT.

DB2 Date Format	Pattern	Example
EUR	DD.MM.YYYY	30.10.1994
ISO	YYYY-MM-DD	1994-10-30
JIS	YYYY-MM-DD	1994-10-30
USA	MM/DD/YYYY	10/30/1994

The Oracle default format of 'DD-MON-YY' is not allowed with DB2. As a result, the gateway local date exit is provided to change the Oracle default date format of 'DD-MON-YY' or 'DD-MON-RR' to the DB2 ISO format of 'YYYY-MM-DD' before passing the date to DB2.

The following example demonstrates the most efficient way to enter and select date values in the twenty-first century:

```
ALTER SESSION SET HS_NLS_DATE_FORMAT = 'YYYY-MM-DD';
INSERT INTO EMP (HIREDATE) VALUES ('2008-07-23');
SELECT * FROM EMP WHERE HIREDATE = '2008-07-23';
UPDATE EMP SET HIREDATE = '2008-07-24'
WHERE HIREDATE = '2008-07-23';
DELETE FROM EMP WHERE HIREDATE = '2008-07-24';
```

14.7.7 Oracle TO_DATE Function

The Oracle TO_DATE function is preprocessed in SQL INSERT, UPDATE, DELETE, and SELECT WHERE clauses. TO_DATE functions in SELECT result lists are not preprocessed.

The TO_DATE function is often needed to provide values to update or compare with date columns. Therefore, the gateway replaces the information included in the TO_DATE clause with an acceptable value before the SQL statement is sent to DB2.

Except for the SELECT result list, all TO_DATE functions are preprocessed and turned into values that are the result of the TO_DATE function. Only TO_DATE(literal) or TO_DATE(:bind_variable) is allowed. Except in SELECT result lists, the TO_DATE(column_name) function format is not supported.

The preprocessing of the Oracle TO_DATE functions into simple values is useful in an INSERT VALUES clause because DB2 does not allow functions in the VALUES clause. In this case, DB2 receives a simple value in the VALUES list. All forms of the TO_DATE function (with one, two, or three operands) are supported.

14.7.8 Performing Numeric Datatype Operations

IBM versions of the DRDA Server perform automatic conversions to the numeric datatype of the destination column (such as integer, double-precision floating point, or decimal). The user has no control over the datatype conversion, and this conversion can be independent of the datatype of the destination column in the database.

For example, if `PRICE` is an integer column of the `PRODUCT` table in an IBM DRDA database, then the update shown in the following example inaccurately sets the price of an ice cream cone to \$1.00 because the IBM DRDA Server automatically converts a floating point to an integer:

```
UPDATE PRODUCT
SET PRICE = 1.50
WHERE PRODUCT_NAME = 'ICE CREAM CONE';
```

Because `PRICE` is an integer, the IBM DRDA Server automatically converts the decimal data value of 1.50 to 1.

14.7.9 Mapping the COUNT Function

The Oracle Database server supports the following four operands for the `COUNT` function:

- `COUNT(*)`
- `COUNT(DISTINCT colname)`
- `COUNT(ALL colname)`
- `COUNT(colname)`

The default is `COUNT(ALL colname)`.

IBM versions of the DRDA Server support only two operands for the `COUNT` function:

- `COUNT(*)`
- `COUNT(DISTINCT colname)`

When an Oracle application issues a `COUNT(colname)` or a `COUNT(ALL colname)`, the gateway translates the request to `COUNT(*)`, which is compliant DRDA Server SQL syntax. `COUNT(*)` includes null values whereas `COUNT(colname)` or `COUNT(ALL colname)` does not.

To prevent null rows from being counted, when using `COUNT(colname)` or `COUNT(ALL colname)` against a DRDA Server, use the following commands:

```
SELECT COUNT(colname) FROM EMP WHERE colname IS NOT NULL
or
```

```
SELECT COUNT(ALL colname) FROM EMP WHERE colname IS NOT NULL
```

14.7.10 Performing Zoned Decimal Operations

A zoned decimal field is described as packed decimal on an Oracle Database server. However, an Oracle application such as a Pro*C program can insert into a zoned decimal column using any supported Oracle numeric datatype. The gateway converts this number into the most suitable datatype. Data can be fetched from a DRDA database into any Oracle datatype, provided that it does not result in a loss of information.

14.8 Passing Native SQL Statements through the Gateway

The passthrough SQL feature allows an application developer to send a SQL statement directly to the DRDA Server without the statement being interpreted by the Oracle Database server. DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE SQL passthrough statements that are supported by the gateway are limited to nonqueries (INSERT, UPDATE, DELETE, and DDL statements) and cannot contain bind variables. The gateway can run native SQL statements using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE.

DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE is a built-in gateway function. This function receives one input argument and returns the number of rows affected by the SQL statement. For DDL statements, the function returns zero.

DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE are reserved names of the gateway and are used specifically for running native SQL.

This release of Oracle Transparent Gateway for DRDA enables retrieval of result sets from queries issued with passthrough. The syntax is different from the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function. Refer to ["Retrieving Results Sets Through Passthrough"](#) on page 14-29 for more information.

14.8.1 Using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE

Using the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function

To run a passthrough SQL statement using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE, use the following syntax:

```
number_of_rows = DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink ('native_DRDA_sql');  
where:
```

`number_of_rows` is a variable that is assigned the number of rows affected by the passthrough SQL completion. For DDL statements, a zero is returned for the number of rows affected.

`dblink` is the name of the database link used to access the gateway.

`native_DRDA_sql` is a valid nonquery SQL statement (except CONNECT, COMMIT, and ROLLBACK). The statement cannot contain bind variables. Native SQL statements that cannot be dynamically prepared are rejected by the DRDA Server. The SQL statement passed by the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function must be a character string. For more information regarding valid SQL statements, refer to the SQL Reference for the particular DRDA Server.

14.8.1.1 Examples

1. Insert a row into a DB2 table using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE:

```
DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink
    ('INSERT INTO SCOTT.DEPT VALUES (10, ''PURCHASING'', ''PHOENIX'')');
END;
/
```

2. Create a table in DB2 using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE:

```
DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink
    ('CREATE TABLE MYTABLE (COL1 INTEGER, COL2 INTEGER, COL3 CHAR(14),
    COL4 VARCHAR(13))');
END;
/
```

14.8.2 Retrieving Results Sets Through Passthrough

Oracle Transparent Gateway for DRDA provides a facility to retrieve results sets from a SELECT SQL statement entered through passthrough. Refer to *Oracle Database Heterogeneous Connectivity Administrator's Guide* for additional information.

14.8.2.1 Example

```
DECLARE
    CRS binary_integer;
    RET binary_integer;
    VAL VARCHAR2(10)
BEGIN
    CRS:=DBMS_HS_PASSTHROUGH.OPEN_CURSOR@gtwlink;
    DBMS_HS_PASSTHROUGH.PARSE@gtwlink(CRS, 'SELECT NAME FROM PT_TABLE');
    BEGIN
        RET:=0;
        WHILE (TRUE)
        LOOP
            RET:=DBMS_HS_PASSTHROUGH.FETCH_ROW@gtwlink(CRS, FALSE);
            DBMS_HS_PASSTHROUGH.GET_VALUE@gtwlink(CRS, 1, VAL);
            INSERT INTO PT_TABLE_LOCAL VALUES(VAL);
        END LOOP;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            BEGIN
                DBMS_OUTPUT.PUT_LINE('END OF FETCH');
                DBMS_HS_PASSTHROUGH.CLOSE_CURSOR@gtwlink(CRS);
            END;
    END;
END;
/
```

14.9 Oracle Data Dictionary Emulation on a DRDA Server

The gateway optionally augments the DRDA database catalogs with data dictionary views modeled after the Oracle data dictionary. These views are based on the dictionary tables in the DRDA database, presenting that catalog information in views familiar to Oracle users. The views created during the installation of the gateway automatically limit the data dictionary information presented to each user based on the privileges of that user.

14.9.1 Using the Gateway Data Dictionary

The gateway data dictionary views provide users with an Oracle-like interface to the contents and use of the DRDA database. Some of these views are required by Oracle products. The gateway supports the DB2/OS390, DB2/400, and DB2/UDB catalog views. DB2/VM catalog views are not available.

You can query the gateway data dictionary views to see the objects in the DRDA database and to determine the authorized users of the DRDA database. Many Oracle catalog views are supported by the Oracle Transparent Gateway for DRDA. Refer to [Appendix A](#) for descriptions of Oracle DB2 catalog views. These views are completely compatible with the gateway.

14.9.2 Using the DRDA Catalog

Each DRDA database has its own catalog tables and views, which you might find useful. Refer to the appropriate IBM documentation for descriptions of these catalogs.

14.10 Defining the Number of DRDA Cursors

You can define any number of cursors depending on your application requirements. Oracle Corporation recommends that you use the default value of 100. However, if the default is not appropriate for your application, there are two points to consider when defining the number of cursors for your installation:

1. Each cursor requires an additional amount of storage and additional management.
2. If you change `DRDA_PACKAGE_SECTIONS`, you must rebind the package.

The parameter `DRDA_PACKAGE_SECTIONS` is specific to the DRDA package. This parameter defines the number of sections (open cursors at the IBM database). Refer to [Appendix C, "DRDA-Specific Parameters"](#) for more information about setting the `DRDA_PACKAGE_SECTIONS` parameter.

Security Considerations

The gateway architecture involves multiple computer systems that have distinct security capabilities and limitations. This chapter provides information for planning and implementing your security system.

This chapter provides information that is specific to this release of the Oracle Transparent Gateway for DRDA. It includes the following sections:

- [Security Overview](#) on page 15-2
- [Authenticating Application Logons](#) on page 15-2
- [Defining and Controlling Database Links](#) on page 15-3
- [Processing Inbound Connections](#) on page 15-3
- [Passwords in the Gateway Initialization File](#) on page 15-5
- [Using the g4drpwd Utility](#) on page 15-6

15.1 Security Overview

When you connect several different systems, generally the system with the strictest security requirements dictates and rules the system.

Gateway security involves two groups:

- Users and applications that are permitted access to a given gateway instance and DRDA database server
- Server database objects that users and applications are able to query and update

You can control access in the gateway architecture at several points. Control over database object access is provided by each DRDA database server with GRANTS and related native authorization mechanisms based on user ID.

When the gateway is involved in a SQL request, security mechanisms are in effect for each DRDA system component encountered by the gateway. The first system component encountered is the application tool or 3GL program. The last system component encountered is the DRDA database.

15.2 Authenticating Application Logons

An application must connect to an Oracle integrating server before using the gateway. The type of logon authentication that you use determines the resulting Oracle user ID and can affect gateway operation. There are two basic types of authentication:

- Oracle authentication

With Oracle authentication, each Oracle user ID has a password known to the Oracle Database server. When an application connects to the server, it supplies a user ID and password. The Oracle Database server confirms that the user ID exists and that the password matches the one kept in the database.

- Operating system authentication

With operating system authentication, the server's underlying operating system is responsible for authentication. An Oracle user ID that is created with the IDENTIFIED EXTERNALLY attribute, instead of a password, is accessed with operating system authentication. To log into such a user ID, the application supplies a forward slash (/) for a user ID and does not supply a password.

To perform operating system authentication, the server determines the requester's operating system user ID, optionally adds a fixed prefix to it, and uses the result as the Oracle user ID. The server confirms that the user ID exists and is IDENTIFIED EXTERNALLY, but no password checking is done. The underlying assumption is that users were authenticated when they logged into the operating system.

Operating system authentication is not available on all platforms and is not available in some Oracle Net (client-server) and multi-threaded server configurations. Refer to the *Oracle Database Installation Guide 10g for UNIX Systems* and Oracle Net documentation to determine the availability of this feature.

For more information about authenticating application logons, refer to the *Oracle Database Reference*.

15.3 Defining and Controlling Database Links

The information here is specific to the gateway. For additional information on database links, refer to the *Oracle Database Reference*.

15.3.1 Link Accessibility

The first point of control for a database link is simply whether it is accessible to a given user. A public database link can be used by any user ID. A private database link is usable only by the user who created it. The server makes no distinction regarding the type of use (such as read-only versus update or write) or which remote objects can be accessed. These distinctions are the responsibility of the DRDA database that is accessed.

15.3.2 Links and CONNECT Clauses

The CONNECT clause is another security-related attribute of a database link. You can use the CONNECT clause to specify an explicit user ID and password, which can differ from the user's Oracle user ID and password. This CONNECT user ID and password combination is sent to the gateway when the database link connection is first opened. Depending on gateway options, the gateway might send that user ID and password to the DRDA Server for it to validate.

If a database link is created without a CONNECT clause, then the user's Oracle user ID and password are sent to the gateway when the connection is opened. If the user logs into the Oracle integrating server with operating system authentication, then the gateway receives no user ID or password from the Oracle integrating server. In this case, user ID mapping facilities at the DRDA Server can be used to make such a connection possible if all users on the same host can use the same DRDA database user ID.

15.4 TCP/IP Security

TCP/IP does not have any additional configurable security mechanism. The gateway supports a validation mechanism which requires a user ID and a valid password. The security information is passed to the DRDA Server for validation. This type of validation is equivalent to the "SNA Security Option SECURITY=PROGRAM". Refer to the discussion of this option in [Chapter 6](#), [Chapter 7](#), [Chapter 8](#), or [Chapter 9](#). The difference between the two methods is that in the SNA configuration, the security validation is performed by the SNA network facilities, while in the TCP/IP configuration, the DRDA Server manually performs the validation.

15.5 Processing Inbound Connections

Current DRDA Servers provide options for manipulating the security conduct of an inbound (client) DRDA session request. Refer to the appropriate IBM documentation for detailed information about the security options discussed in this section. Refer to ["Documentation Requirements"](#) on page 3-4, for a list of IBM documentation.

15.5.1 User ID Mapping

The most useful DRDA Server security capability is user ID mapping. User ID mapping refers to changing the user ID associated with an incoming DRDA request to some other user ID known to that server. This is a useful feature if your installation does not have a uniform user ID structure across all systems and databases.

15.5.1.1 DB2/OS390

The DB2 DDF Communication Database (CDB) stores inbound DRDA session security options.

These tables, pertinent to inbound sessions, have a role in security processing:

- **SYSIBM.LUNAMES table**

The SYSIBM.LUNAMES table controls inbound security conduct on an SNA LU basis, affecting all DRDA connections from a particular HP9000 host system. This table also controls whether inbound connection user IDs are subject to translation or mapping.

- **SYSIBM.SYSUSERNAMES table**

When translation is used, rows in the SYSIBM.SYSUSERNAMES table specify translated user IDs by LU name and inbound user ID. Default entries that pertain to all LUs and to all inbound user IDs can be made in both tables. The mapping table can also be used simply to indicate which inbound user IDs are permitted from a particular LU or from all LUs, whether or not they are mapped.

This implementation provides a flexible mapping structure. You can specify that all connections from a particular LU use a single DB2 user ID, or that a particular inbound user ID always be mapped to a particular DB2 user ID regardless of origin. A SYSUSERNAMES entry with blank LU name and inbound user ID can designate a single default DB2 user ID for all connections unless a more specific entry, by LU name, user ID, or both, exists.

The CDB tables can be updated by a user with update authority using a SQL tool such as the DB2 SPUFI utility. For example, most database administrators, systems programmers, and security officers can update CDB tables. The DB2 DDF component must be stopped and restarted for CDB changes to take effect.

The DB2 non-DRDA-specific security features are also involved in DRDA connections. User IDs are subject to normal DB2 or SAF/RACF validation in addition to connection or sign-on exit processing. Passwords are also subject to validation. After the connection is established, all normal authorizations or GRANTs associated with the user ID are in effect. The user ID must have execute authority on the gateway DRDA package to process any SQL statements.

15.5.1.2 DB2/VM

Under VM, DRDA sessions are managed by APPC VTAM Support (AVS), which runs as a disconnected GCS virtual machine. AVS retrieves incoming APPC connection requests (both DRDA and non-DRDA) and routes the connection to an appropriate server virtual machine.

AVS user ID mapping is controlled by internal AVS data structures that are updated with the AGW ADD USERID and AGW DELETE USERID commands.

A user ID mapping entry converts the inbound user ID before making the DB2/VM connection. The user ID mapping consists of:

- Originating LU name
- Inbound user ID
- The new user ID

You can create default entries that apply to any LU name and to any inbound user ID, and an entry can indicate that the inbound user ID is to be used without mapping.

AVS user ID mapping is functionally similar to the DB2 user ID translation mechanism and can be used to work around a variety of incongruities among user ids on different systems and databases.

After any indicated user ID mapping has been done, inbound DRDA connection requests are forwarded to the specified DB2/VM server machine. DB2/VM confirms only that the user ID has CONNECT authority and, if so, that the connection is complete. At this point, the application's access to DB2/VM objects is controlled by the normal authorities and GRANTs for the connected user ID. The user ID must have execute authority on the gateway DRDA package to process any SQL statements.

15.5.1.3 DB2/400

DB2/400 does not provide a user ID mapping capability comparable to that in DB2/OS390 and DB2/VM. Normally, the user ID in an incoming DRDA connection request must be a valid user ID on that AS/400.

The AS/400 subsystem communications entry for the gateway should specify that the gateway is not a secure location and should include a default user ID of *NONE.

After the application has completed the DRDA connection to the AS/400, it is subject to all authorities and GRANTs associated with the user ID in use.

The user ID must have execute authority on the gateway DRDA package to execute any SQL statements.

15.5.1.4 DB2/Universal Database

DB2/Universal Database (DB2/UDB) does not provide a user ID mapping capability comparable to that in DB2/OS390 and DB2/VM. Normally, the user ID in an incoming DRDA connection request must be a valid user ID on the DB2/UDB host.

After the application has completed the DRDA connection to the DB2 host, it is subject to all authorities and GRANTs associated with the user ID in use. The user ID must have execute authority on the gateway DRDA package to execute any SQL statements.

15.6 Passwords in the Gateway Initialization File

The gateway uses userids and passwords to access the information in the remote database on the DRDA Server. Some userids and passwords must be defined in the Gateway Initialization File to handle functions such as resource recovery. Refer to the parameters DRDA_RECOVERY_USERID and DRDA_RECOVERY_PASSWORD in [Appendix C](#) as examples. In the current security conscious environment, having plain-text passwords that are accessible in the Initialization File is deemed insecure. A new encryption feature has been added to the gateway to help make this more secure. The **g4drpwd** utility can be used to encrypt passwords that would normally be stored in the Initialization File. Using this feature is optional, but it is highly recommended by Oracle Corporation. With this feature, passwords are no longer stored in the Initialization File, but instead are stored in a password file in an encrypted form, thus making the information more secure. Read the next section to learn how to use this feature.

15.7 Using the g4drpwd Utility

The **g4drpwd** utility is used to encrypt passwords that would normally be stored in the Gateway Initialization File. The utility works by reading the Initialization File and looking for parameters with a special value. The specific value is the asterisk ("*"). This designates that the value of this parameter is stored in an encrypted form in another file. The following is a sample section of the Initialization File with this value ("*"):

```
# DRDA_RECOVERY_PASSWORD: Default: none, must be a valid MVS password.
# The recovery user connects to the IBM database if a distributed transaction
# is in doubt.
DRDA_RECOVERY_PASSWORD=*
```

The Initialization File is first edited to set the value of the parameter to "*" (asterisk). Then the **g4drpwd** utility is run, specifying the gateway SID on the command line. The utility will read the Initialization File and will prompt the user to enter the values that are to be encrypted.

The syntax of the command is: `g4drpwd [gateway_sid]`

Where `[gateway_sid]` is the SID of the gateway.

The following is an example run, assuming a gateway sid of **DB2**:

```
$ g4drpwd DB2
ORACLE Gateway Password Utility (tg4drda)
Constructing password file for Gateway SID DB2
Enter the value for DRDA_RECOVERY_PASSWORD
ORADDRDA
```

In the example above, the parameter "DRDA_RECOVERY_PASSWORD" is identified as requiring encryption. The user enters the value (in other words, "ORADDRDA") and presses enter. If more parameters require encryption, then they will be prompted for in turn. The encrypted data is stored in the **tg4drda/admin** directory.

Note: It is important that the **ORACLE_HOME** environmental variable be pointing to the correct gateway home to ensure that the correct Gateway Initialization File is read.

Migration and Coexistence with Existing Gateways

Migration to new instances of Oracle Transparent Gateway for DRDA from an existing installation is straightforward, provided some guidelines are followed. This chapter provides information to make these new installations as easy as possible.

This chapter provides information that is specific to this release of the Oracle Transparent Gateway for DRDA, including the following sections:

- [Migrating Existing V4, V8, or V9 Gateway Instances to New Release](#) on page 16-2
- [Backout Considerations When Migrating to New Releases](#) on page 16-2
- [New and Changed Parameters When Migrating to Release 10](#) on page 16-3
- [DRDA Server Considerations](#) on page 16-4
- [Oracle Net Considerations](#) on page 16-4

16.1 Migrating Existing V4, V8, or V9 Gateway Instances to New Release

Migration is the process of transforming an installed version of an Oracle Database into a later version (Compare this with upgrading). For example, transforming an Oracle8i database into an Oracle9i database is migrating the database. This transformation generally involves running the Oracle migrate (MIG) utility to modify Oracle Database control file structures from the format of one version to the format of another version.

Upgrading is the process of transforming an Oracle Database from an installed release into a later release of the same version. For example, transforming patch release 8.0.3 into patch release 8.0.4 is upgrading.

16.1.1 Step 1: Install the new Release

Install the new release of the Gateway in a separate directory, as outlined in [Chapter 4](#), "Installing the Gateway".

Caution: Do not install the Gateway over a previously existing Gateway installation. Doing so will corrupt that existing installation.

16.1.2 Step 2: Transferring *initsid.gtwboot* Gateway Boot Initialization parameters.

In previous releases, the gateway used two gateway initialization files (*initsid.gtwboot* and *initsid.ora*), or it used a Startup Shell Script (*drdaDB2.sh*) and one initialization file (*initsid.ora*). In this release, all parameters have been migrated into a single gateway initialization file: *initsid.ora*. Migrating a previous release involves copying the parameters from the *initsid.gtwboot* or Startup Shell Script into the *initsid.ora*. The format of the parameters can be found in [Appendix C](#), "DRDA-Specific Parameters".

16.1.3 Step 3: Transferring *initsid.ora* Gateway Initialization File parameters.

Copy the *initsid.ora* from the old Gateway instance to the new instance. The parameters in the *initsid.ora* Gateway Initialization File have changed format. Refer to "Gateway Initialization File Parameters" on page C-2 in [Appendix C](#), "DRDA-Specific Parameters".

16.2 Backout Considerations When Migrating to New Releases

During the migration from older version 4, version 8, or version 9 gateway instances to the latest Oracle Database 10g release, if problems are encountered, then it is always possible to revert to the previous version. Assuming a working version 4 gateway instance exists, simply change the TNSNAMES.ORA entries from using the Oracle Database 10g gateway instance to the older version 4 instance. Remember to remove the "(HS=)" entry from the SQL*Net connect definition.

Oracle recommends that when you are installing a new release of the gateway and upgrading existing instances, that you keep the old gateway home and instance configurations intact and operational in case you have problems with the upgrade. This will help ensure minimal downtime between changes to different gateway instances.

16.3 New and Changed Parameters When Migrating to Release 10

This release of the Oracle Transparent Gateway for DRDA introduces new and changed initialization parameters if you are migrating from a Version 4, version 8, or version 9 gateway to the Oracle Database 10g gateway.

16.3.1 New Parameters

The following section lists new parameters relevant to migration from Version 4 gateways.

16.3.1.1 New Gateway Initialization File Parameters

Parameters introduced in this release of the gateway, listed in the following table, may be added to the Gateway Initialization File:

- DRDA_CACHE_TABLE_DESC
- DRDA_GRAPHIC_LIT_CHECK
- DRDA_GRAPHIC_PAD_SIZE
- DRDA_GRAPHIC_TO_MBCS
- DRDA_MBCS_TO_GRAPHIC

16.3.2 Parameters That Have Been Changed in Usage

The usage of the following parameter has changed with version 9 of the gateway:

- DRDA_CONNECT_PARM

16.3.3 Parameters That Have Been Renamed

The following table presents a list of parameters that have been renamed with version 9 of the gateway, and their corresponding old names. Refer to *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more detailed information about these parameters.

New Name	Old Name
HS_COMMIT_STRENGTH_POINT	COMMIT_STRENGTH_POINT
HS_DB_DOMAIN	DB_DOMAIN
HS_DB_INTERNAL_NAME	DB_INTERNAL_NAME
HS_DB_NAME	DB_NAME
HS_DESCRIBE_CACHE_HWM	DESCRIBE_CACHE_HWM
HS_LANGUAGE	LANGUAGE
HS_NLS_DATE_FORMAT	NLS_DATE_FORMAT
HS_NLS_DATE_LANGUAGE	NLS_DATE_LANGUAGE
HS_OPEN_CURSORS	OPEN_CURSORS
HS_ROWID_CACHE_SIZE	ROWID_CACHE_SIZE

16.3.4 Obsolete Parameters

The following parameters are now obsolete. Please remove them from your configuration files:

- MODE
- SERVER_PATH
- DRDA_OVERRIDE_FROM_CODEPAGE
- DRDA_OVERRIDE_TO_CODEPAGE
- ERROR_LOGGING
- ERROR_REPORTING
- ERRORTAG
- GATEWAY_SID
- GROUP_BY_OFF
- GTWDEBUG
- INCREMENT_CURSORS

16.4 DRDA Server Considerations

Part of the normal installation for the gateway involves binding a package and (as an option) installing data dictionary views on the DRDA Server. This release of the gateway (10.1.0.2.0) is compatible with version 4, version 8, and version 9 packages that have been previously bound. The data dictionary views, however, have changed with this release. If you plan to utilize the data dictionary views that are provided by the gateway, then you must migrate to the new views. Oracle Corporation recommends that you install the new views as outlined in [Chapter 12, "Configuring the Gateway"](#). If you have changed certain DRDA parameters of the gateway initialization parameters as a result of the migration, then a rebind of the package will be required.

16.5 Oracle Net Considerations

The Gateway uses the Heterogeneous Services (HS) facilities of Oracle and Oracle Net. As such, gateway service name entries in the **tnsnames.ora** need a slight modification to tell Oracle Net that the gateway will be using the HS facilities. Refer to ["Configuring Oracle Net"](#) on page 11-4 for detailed information.

Error Messages, Diagnosis, and Reporting

This chapter provides information about error messages and error codes. This data is specific to this release of the Oracle Transparent Gateway for DRDA, including the following sections:

- [Interpreting Gateway Error Messages](#) on page 17-2
- [Mapped Errors](#) on page 17-4
- [Gateway Error Codes](#) on page 17-5
- [SQL Tracing and the Gateway](#) on page 17-6

17.1 Interpreting Gateway Error Messages

The gateway architecture involves a number of separate components. Any component might detect and report an error condition while processing SQL statements that refer to one or more DRDA database tables. This means that error situations can be complex, involving error codes and supporting data from multiple components. In all cases, however, the application ultimately receives a single Oracle error number or return code upon which to act.

Because most gateway messages exceed the 70 character message area in the Oracle SQLCA, the programmatic interfaces and Oracle Call Interfaces that you use to access data through the gateway should use SQLGLM or OERHMS to view the entire text of messages. Refer to the programmer's guide to the Oracle precompilers for additional information about SQLGLM, and see the *Oracle C++ Call Interface Programmer's Guide* for additional information about OERHMS. The error messages listed below apply to both TCP/IP and SNA networking communications products on the gateway.

Error conditions encountered when using the gateway can originate from many sources:

- Errors detected by the Oracle integrating server
- Errors detected by the gateway
- Errors detected in the DRDA software, either on the requestor or server side
- Communication errors
- Errors detected by the server database

17.1.1 Errors Detected by the Oracle Integrating Server

Errors detected by the Oracle integrating server are reported back to the application or tool with the standard "ORA-" type message. Refer to *Oracle Database Error Messages* for descriptions of these errors. For example, the following error message occurs when an undefined database link name is specified:

```
ORA-02019: connection description for remote database not found
```

Errors in the ORA-9100 to ORA-9199 range are reserved for the generic gateway layer (components of the gateway that are not specific to DRDA). Messages in this range are documented in *Oracle Database Error Messages*.

17.1.2 Errors Detected by the Gateway

Errors detected by the generic gateway are prefixed with "HGO-" and are documented in *Oracle Database Error Messages*.

An example error message is:

```
HGO-00706: HGO: Missing equal sign for parameter in initialization file.
```

17.1.3 Errors Detected in the DRDA Software

Errors detected in the DRDA gateway, on the requestor or server side, are usually reported with error ORA-28500, followed by a gateway-specific expanded error message. There are two return codes reported in the expanded message:

- `drc` specifies DRDA-specific errors which are documented in "[Gateway Error Codes](#)" on page 17-5.
- `grc` specifies generic gateway errors detected in the DRDA layer. These errors are documented in the *Oracle Database Error Messages*.

Note: Error code ORA-28500 was error code ORA-09100 prior to gateway version 8. Error code ORA-28501 was error code ORA-09101 prior to gateway version 8.

The values in parentheses that follow the `drc` values are used for debugging by Oracle Support Services. The `errp` field indicates the program (requestor or server) that detected the error. If present, `errmc` lists any error tokens.

For example, the following error message is returned when the database name specified (`XNAME`) with the `DRDA_REMOTE_NAME` parameter in the `initsid.ora` file is not defined at the DRDA Server:

```
ORA-28500: connection from ORACLE to a non-Oracle system returned the message:
TG4DRDA v10.1.0.2.0 grc=0, drc=-30061 (839C,0000), errp=GDJRFS2E
errmc=XNAME
```

17.1.4 Communication Errors

Communication errors are reported with an ORA-28501 followed by a gateway-specific expanded error message with `drc=-30080` (SNA CPI-C error) or `drc=-30081` (lost session). `errmc` indicates which CPI-C routine encounters the error, followed by the CPI-C error code and error number.

For example, the following error message is returned when there is a failure to establish a session because `DRDA_CONNECT_PARM` in the `initsid.ora` file specifies a Side Information Profile that is not defined:

```
ORA-28501: communication error on heterogeneous database link
TG4DRDA v10.1.0.2.0 grc=0, drc=-30081 (839C,0001), errp= file or directory(2)
errmc=Initialize_Conversation (CMINIT) CM_PROGRAM_PARAMETER_CHECK(24) No such >
file or directory(2)
```

Refer to the appropriate host operating system, or SNA server documentation for more information.

17.1.5 Errors Detected by the Server Database

Errors detected by the server database are reported with an ORA-28500 followed by a gateway-specific expanded error message with `drc=-777` (`sqlcode` follows.) This is followed by another error message line that contains the `sqlcode`, `sqlstate`, `errd` (error array), and `errmc` (error tokens) returned from the DRDA Server database. Refer to IBM documentation for the specific database being used. Also refer to [Mapped Errors](#) in this chapter for some SQL errors that get translated.

Note: Error code ORA-28500 was error code ORA-09100 prior to gateway version 8. Error code ORA-28501 was error code ORA-09101 prior to gateway version 8.

For example, the following error message indicates that the DRDA Server database did not recognize the collection ID or package name specified with the DRDA_PACKAGE_COLLID or DRDA_PACKAGE_NAME parameters in the **initsid.ora** file:

```
ORA-28500: connection from ORACLE to a non-Oracle system returned the message:
TG4DRDA v10.1.0.2.0 grc=0, drc=-30020 (839C,0000), errp=GDJMRCM
sqlcode=-805, sqlstate=51002, errd=FFFFFFF9C,0,0,FFFFFFF,0,0
errmc=124c
```

17.2 Mapped Errors

Some SQL errors are returned from the DRDA Server database and are translated to an Oracle error code. This is needed when the Oracle instance or gateway provides special handling of an error condition. The following table lists the mapped sqlstate error numbers, descriptions, and their corresponding Oracle error codes:

Table 17–1 Mapped sqlstate Errors

Description	sqlstate error	Oracle error
No rows selected	02000	0
Unique index constraint violated	23505	ORA-00001
Object does not exist	52004 or 42704	ORA-00942
Object name too long (more than 18 characters), and therefore object does not exist	54003 or 42622	ORA-00942
Insufficient privileges	42501	ORA-01031
Invalid CCSID (unimplemented character set conversion)	22522	ORA-01460
Invalid username/password; logon denied	N/A	ORA-01017
Divide by zero error	01519 or 01564	ORA-01476

The following is an example of a translated "object does not exist" error:

```
ORA-00942: table or view does not exist
TG4DRDA v10.1.0.2.0 grc=0, drc=-942 (839C,0001), errp=DSNXEDST
sqlcode=-204, sqlstate=52004, errd=32,0,0,FFFFFFF,0,0
errmc=AJONES.CXDCX
```

17.3 Gateway Error Codes

Listed below are the common Oracle Transparent Gateway for DRDA error codes that appear in the `drc=` field of the expanded error messages. If you obtain a `drc` value that does not appear here, contact Oracle Support Services.

-700 Invalid ORA_MAX_DATE specified

Cause: An invalid value was specified for ORA_MAX_DATE in the `initsid.ora` file.

Action: Correct the value of ORA_MAX_DATE. Correct format is ORA_MAX_DATE=YYYY-MM-DD, where MM is in the range of 1 to 12, and DD is in the range of 1 to 31 (and must be valid for the month).

-701 Default CCSID value not supported

Cause: The value specified for DRDA_DEFAULT_CCSID in the `initsid.ora` file is not supported by the Oracle Transparent Gateway for DRDA.

Action: Refer to [Appendix D, "National Language Support"](#), for a list of supported DRDA Server character sets.

-702 Application Host (bind) variable exceeds 32K

Cause: An application program specified a host variable with length greater than the DRDA allowed maximum of 32K.

Action: The application must be modified to take into account DRDA limits.

-703 Local Character set not supported

Cause: The character set specified for the LANGUAGE parameter in the `initsid.ora` file is not supported.

Action: Refer to [Appendix D, "National Language Support"](#), for a list of supported character sets.

-704 User id length greater than maximum

Cause: The user ID being used for the allocation of an APPC conversion by the gateway is longer than 8 characters.

Action: A user ID of length of 8 or less must be used. Refer to [Chapter 15, "Security Considerations"](#), for a discussion of user IDs.

-705 Password length greater than maximum

Cause: The password being used for the allocation of an APPC conversion by the gateway is longer than 8 characters.

Action: A password of length of 8 or less must be used. Refer to [Chapter 15, "Security Considerations"](#), for a discussion of passwords.

-777 DRDA Server RDBMS (SQL) Error

Cause: Server database detected an application-level SQL error.

Action: Refer to ["Interpreting Gateway Error Messages"](#) on page 17-2. `sqlcode` and `sqlstate` indicate host database error. Use this information to fix your application.

-30060 Invalid Userid/Password (DRDA Server RDBMS Authorization)

Cause: You have used a user ID/password that is not acceptable to the DRDA Server database.

Action: Refer to [Chapter 15, "Security Considerations"](#), for user ID/password considerations.

-30061 RDB not found

Cause: The remote database specified with the DRDA_REMOTE_DB_NAME parameter is not a valid database at the DRDA Server.

Action: Correct the value of the DRDA_REMOTE_DB_NAME parameter in the `initsid.ora` file.

-30080 Communication Error

Cause: The gateway encountered a CPI-C communication error.

Action: Retry processing that received error. If it persists, then refer to ["Interpreting Gateway Error Messages"](#) on page 17-2 and report to your system administrator.

-30081 Communication Error - lost session

Cause: The current DRDA CPI-C session was disconnected.

Action: Retry processing that received error. If it persists, then refer to ["Interpreting Gateway Error Messages"](#) on page 17-2 and report it to your system administrator.

17.4 SQL Tracing and the Gateway

When developing applications it is often useful to be able to see the exact SQL statements that are being passed through the Gateway. This section describes setting appropriate trace parameters and setting up the debug Gateway.

17.4.1 SQL Tracing in the Oracle Database

The Oracle Database server has a command for capturing the SQL which is actually sent to the gateway. This command is called EXPLAIN PLAN. EXPLAIN PLAN is used to determine the execution plan that Oracle follows to execute a specified SQL statement. This command inserts a row (describing each step of the execution plan) into a specified table. If you are using cost-based optimization, this command also determines the cost of executing the statement. The syntax of the command is:

```
EXPLAIN PLAN [ SET STATEMENT_ID = 'text' ]  
[ INTO [schema.]table[@dblink] ] FOR statement
```

For detailed information on this command, refer to the *Oracle Database SQL Reference*.

Note: In most cases, EXPLAIN PLAN should be sufficient to extract the SQL which is actually sent to the gateway, and thus sent to the DRDA Server. However, certain SQL statement forms have post-processing performed on them in the gateway. The next section will describe setting up SQL Tracing in the gateway.

17.4.2 SQL Tracing in the Gateway

The production gateway does not have tracing built into it for the purpose of enhancing its speed. The product ships with a debug library that can be used to build a debug gateway for the purposes of tracing and debugging of applications.

First, login as the Admin user ID of the gateway and setup the environment:

```
$ su - <Gateway-Admin-User>
```

Next, build the debug gateway:

```
$ cd $ORACLE_HOME/tg4drda/lib
$ make -f tg4drda.mk ORACLE_HOME=your_oracle_home g4drsrvd
```

Note: ORACLE_HOME is the location of your gateway installation. That is to say, "*your_oracle_home*" is the ORACLE_HOME of the gateway, as set in "[Step 3: Set the ORACLE_HOME environment variable](#)" on page 4-4 in [Chapter 4, "Installing the Gateway"](#).

This will create the debug gateway and store it in **\$ORACLE_HOME/bin/g4drsrvd**. Next, change the **listener.ora** to invoke the debug gateway. Using our example from [Appendix B, "Sample Files"](#), change the PROGRAM parameter to specify the debug program name:

```
(SID_DESC=
  (SID_NAME=drdahoal)
  (ORACLE_HOME=/oracle/tg4drda/10.1.0)
  (PROGRAM=g4drsrvd)
```

The listener will need to be reloaded for this change to take effect. Next, edit the gateway initialization file and add the following parameters:

TRACE_LEVEL and

ORACLE_DRDA_TCTL

You may optionally add the LOG_DESTINATION parameter, but it is not required.

The following is a fragment of a Gateway Initialization File with the parameters set:

```
#
TRACE_LEVEL=255
ORACLE_DRDA_TCTL=debug.tctl
#
```

The above example will give full tracing of both gateway and DRDA tracing. In many cases, only the gateway tracing is desirable. To obtain only gateway tracing, remove (or comment out) the "ORACLE_DRDA_TCTL" parameter.

If you specify a LOG_DESTINATION, you may specify just the file name (for example, "drda.trc"), in which case the log will be written to the gateway's log directory (**\$ORACLE_HOME/tg4drda/log**). Or you may specify a fully qualified path name. If you do not specify a LOG_DESTINATION, a unique log file in a default format will be generated.

The logfile name will be of the form:

```
gatewaysid_pid.trc
```

where:

`gatewaysid` is the SID of the gateway. The value of this is determined by the setting of the FDS_INSTANCE parameter, and `pid` is the process identifier (PID) of the gateway process.

An example log file name would be:

```
drdahoa1_3875.trc
```

When searching for the SQL statements which are passed to the DRDA Server, look for the strings '*** HGAPARS ***' and '*** HGAXMSQL ***'. The string after HGAPARS will be the incoming statement from the Oracle Database 10g RDBMS. The string after HGAXMSQL will be the outgoing statement after any date substitution is done. This is the actual SQL statement which will be given to the DRDA Server.

When you are done developing your application, revert the **PROGRAM=** value in the **listener.ora** to its previous value and reload the listener in order to use the production gateway again. You should also comment out the trace parameters in the Gateway Initialization Files.

Oracle DB2 Data Dictionary Views

This appendix includes the Oracle Transparent Gateway for DRDA data dictionary views accessible to all users of an Oracle Database server. Most views can be accessed by any user with SELECT privileges for DB2 catalog tables.

N/A is used in the following tables to mean that the column is not valid for the gateway.

This appendix contains the following sections:

- [Supported Views](#) on page A-2
- [Data Dictionary View Tables](#) on page A-3

A.1 Supported Views

The following is a list of Oracle data dictionary views that are supported by the gateway for DB2/OS390, DB2/400, and DB2/UDB DRDA Servers. This release of the gateway does not have data dictionary view support for DB2/VM servers.

- ALL_CATALOG
- ALL_COL_COMMENTS
- ALL_CONS_COLUMNS
- ALL_CONSTRAINTS
- ALL_INDEXES
- ALL_IND_COLUMNS
- ALL_OBJECTS
- ALL_SYNONYMS
- ALL_TAB_COMMENTS
- ALL_TABLES
- ALL_TAB_COLUMNS
- ALL_USERS
- ALL_VIEWS
- COL_PRIVILEGES
- DICTIONARY
- DUAL
- TABLE_PRIVILEGES
- USER_CATALOG
- USER_COL_COMMENTS
- USER_CONSTRAINTS
- USER_CONS_COLUMNS
- USER_INDEXES
- USER_OBJECTS
- USER_SYNONYMS
- USER_TABLES
- USER_TAB_COLUMNS
- USER_TAB_COMMENTS
- USER_USERS
- USER_VIEWS

A.2 Data Dictionary View Tables

The remainder of this chapter contains tables describing data dictionary views. In the following descriptions, all are supported for DB2/OS390 and DB2/400.

A.2.1 ALL_CATALOG

All tables, views, synonyms, and sequence accessible to the user:

column name	description
OWNER	Owner of the object
TABLE_NAME	Name of the object
TABLE_TYPE	Type of object

A.2.2 ALL_COL_COMMENTS

Comments on columns of accessible tables and views:

column name	description
OWNER	Owner of the object
TABLE_NAME	Object name
COLUMN_NAME	Column name
COMMENTS	Comments on column

A.2.3 ALL_CONS_COLUMNS

Information about accessible columns in constraint definitions:

column name	description
OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
TABLE_NAME	Name associated with table with constraint definition
COLUMN_NAME	Name associated with column specified in the constraint definition
POSITION	Original position of column in definition

A.2.4 ALL_CONSTRAINTS

Constraint definitions on accessible tables:

column name	description
OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
CONSTRAINT_TYPE	Type of constraint definition
TABLE_NAME	Name associated with table with constraint definition
SEARCH_CONDITION	Text of search condition for table check
R_OWNER	Owner of table used in referential constraint
R_CONSTRAINT_NAME	Name of unique constraint definition for referenced table
DELETE_RULE	Delete rule for referential constraint
STATUS	Status of constraint
DEFERRABLE	Whether the constraint is deferrable
DEFERRED	Whether the constraint was initially deferred
VALIDATED	Whether all data obeys the constraint
GENERATED	Whether the name of the constraint is user or system generated
BAD	Constraint specifies a century in an ambiguous manner
RELY	Whether an enabled constraint is enforced or unenforced
LAST_CHANGE	When the constraint was last enabled or disabled
INDEX_OWNER	N/A
INDEX_NAME	N/A

A.2.5 ALL_INDEXES

Description of indexes on tables accessible to the user:

column name	description
OWNER	Owner of the index
INDEX_NAME	Name of the index
INDEX_TYPE	Type of index
TABLE_OWNER	Owner of the indexed object
TABLE_NAME	Name of the indexed object
TABLE_TYPE	Type of the indexed object
UNIQUENESS	Uniqueness status of the index
COMPRESSION	N/A
PREFIX_LENGTH	0
TABLESPACE_NAME	Name of the tablespace containing the index
INI_TRANS	N/A
MAX_TRANS	N/A

column name	description
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
PCT_THRESHOLD	Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	Column ID of the last column to be included in index-organized table
FREELISTS	Number of process freelists allocated to this segment
FREELIST_GROUPS	Number of freelist groups allocated to this segment
PCT_FREE	N/A
LOGGING	Logging information
BLEVEL	Depth of the index from its root block to its leaf blocks. A depth of 1 indicates that the root block and the leaf block are the same.
LEAF_BLOCKS	Number of leaf blocks in the index
DISTINCT_KEYS	Number of distinct indexed values. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table.
AVG_LEAF_BLOCKS_PER	N/A
AVG_DATA_BLOCKS_PER	N/A
CLUSTERING_FACTOR	N/A
STATUS	State of the index: VALID
NUM_ROWS	Number of rows in the index
SAMPLE_SIZE	Size of the sample used to analyze the index
LAST_ANALYZED	Date on which this index was most recently analyzed
DEGREE	Number of threads per instance for scanning the index
INSTANCES	Number of instances across which the index is to be scanned
PARTITIONED	Whether this index is partitioned
TEMPORARY	Whether the index is on a temporary table
GENERATED	Whether the name of the index is system generated
SECONDARY	N/A
BUFFER_POOL	Whether the index is a secondary object
USER_STATS	N/A
DURATION	N/A
PCT_DIRECT_ACCESS	N/A
ITYP_OWNER	N/A
ITYP_NAME	N/A
PARAMETERS	N/A

column name	description
GLOBAL_STATS	N/A
DOMIDX_STATUS	N/A
DOMIDX_OPSTATUS	N/A
FUNCIDX_STATUS	N/A
JOIN_INDEX	N/A
IOT_REDUNDANT_PKEY_ELIM	N/A

A.2.6 ALL_IND_COLUMNS

ALL_IND_COLUMNS describes the columns of indexes on all tables that are accessible to the current user.

column names	description
INDEX_OWNER	Owner of the index
INDEX_NAME	Name of the index
TABLE_OWNER	Owner of the table or cluster
TABLE_NAME	Name of the table or cluster
COLUMN_NAME	Column name or attribute of object type column
COLUMN_POSITION	Position of column or attribute within the index
COLUMN_LENGTH	Indexed length of the column
CHAR_LENGTH	Maximum codepoint length of the column
DESCEND	Whether the column is sorted in descending order (Y/N)

A.2.7 ALL_OBJECTS

Objects accessible to the user:

column name	description
OWNER	Owner of the object
OBJECT_NAME	Name of object
SUBOBJECT_NAME	Name of the subobject
OBJECT_ID	Object number of the object
DATA_OBJECT_ID	Dictionary object number of the segment that contains the object
OBJECT_TYPE	Type of object
CREATED	N/A
LAST_DDL_TIME	N/A
TIMESTAMP	N/A
STATUS	State of the object
TEMPORARY	Whether the object is temporary
GENERATED	Was the name of this object system generated?
SECONDARY	N/A

A.2.8 ALL_SYNONYMS

All synonyms accessible to the user:

column name	description
OWNER	Owner of the synonym
SYNONYM_NAME	Name of the synonym
TABLE_OWNER	Owner of the object referenced by the synonym
TABLE_NAME	Name of the object referenced by the synonym
DB_LINK	N/A

A.2.9 ALL_TABLES

Description of tables accessible to the user:

column name	description
OWNER	Owner of the table
TABLE_NAME	Name of the table
TABLESPACE_NAME	Name of the tablespace containing the table
CLUSTER_NAME	N/A
IOT_NAME	Name of the index organized table
PCT_FREE	N/A
PCT_USED	N/A
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
FREELISTS	Number of process freelists allocated to this segment
FREELIST_GROUPS	Number of freelist groups allocated to this segment
LOGGING	Logging attribute
BACKED_UP	N/A
NUM_ROWS	Number of rows in the table
BLOCKS	N/A
EMPTY_BLOCKS	N/A
AVG_SPACE	N/A
CHAIN_CNT	N/A
AVG_ROW_LEN	Average length of a row in the table in bytes
AVG_SPACE_FREELIST_BLOCKS	The average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	The number of blocks on the freelist

column name	description
DEGREE	The number of threads per instance for scanning the table
INSTANCES	The number of instances across which the table is to be scanned
CACHE	Whether the cluster is to be cached in the buffer cache
TABLE_LOCK	Whether table locking is enabled or disabled
SAMPLE_SIZE	Sample size used in analyzing this table
LAST_ANALYZED	Date on which this table was most recently analyzed
PARTITIONED	Indicates whether this table is partitioned
IOT_TYPE	If this is an index organized table
TEMPORARY	Can the current session only see data that it placed in this object itself?
SECONDARY	N/A
NESTED	If the table is a nested table
BUFFER_POOL	The default buffer pool for the object
ROW_MOVEMEN	N/A
GLOBAL_STATS	N/A
USER_STATS	N/A
DURATION	N/A
SKIP_CORRUPT	N/A
MONITORING	N/A
CLUSTER_OWNER	N/A
DEPENDENCIES	N/A
COMPRESSION	N/A

A.2.10 ALL_TAB_COLUMNS

Columns of all tables, views, and clusters accessible to the user:

column name	description
OWNER	Owner of the table or view
TABLE_NAME	Table or view name
COLUMN_NAME	Column name
DATA_TYPE	Datatype of column
DATA_TYPE_MOD	Datatype modifier of the column
DATA_TYPE_OWNER	Owner of the datatype of the column
DATA_LENGTH	Maximum length of the column in bytes
DATA_PRECISION	N/A
DATA_SCALE	Digits to the right of decimal point in a number
NULLABLE	Does the column allow nulls? Value is <i>n</i> if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY key.
COLUMN_ID	Sequence number of the column as created
DEFAULT_LENGTH	N/A
DATA_DEFAULT	N/A
NUM_DISTINCT	Number of distinct values in each column of the table
LOW_VALUE	For tables with more than three rows, the second lowest and second highest values. These statistics are expressed in hexadecimal notation for the internal representation of the first 32 bytes of the values.
HIGH_VALUE	N/A
DENSITY	N/A
NUM_NULLS	The number of nulls in the column
NUM_BUCKETS	The number of buckets in histogram for the column
LAST_ANALYZED	The date on which this column was most recently analyzed
SAMPLE_SIZE	The sample size used in analyzing this column
CHARACTER_SET_NAME	The name of the character set
CHAR_COL_DECL_LENGTH	The length of the character set
GLOBAL_STATS	N/A
USER_STATS	N/A
AVG_COL_LEN	Average length of the column (in bytes)
CHAR_LENGTH	Displays the length of the column in characters
CHAR_USED	N/A

A.2.11 ALL_TAB_COMMENTS

Comments on tables and views accessible to the user:

column name	description
OWNER	Owner of the object
TABLE_NAME	Name of the object
TABLE_TYPE	Type of object
COMMENTS	Comments on the object

A.2.12 ALL_USERS

Information about all users of the database:

column name	description
USERNAME	Name of the user
USER_ID	N/A
CREATED	N/A

A.2.13 ALL_VIEWS

Text of views accessible to the user:

column name	description
OWNER	Owner of the view
VIEW_NAME	Name of the view
TEXT_LENGTH	Length of the view text
TEXT	View text. Only the first row of text is returned, even if multiple rows exist.
TYPE_TEXT_LENGTH	Length of the type clause of the typed view
TYPE_TEXT	Type clause of the typed view
OID_TEXT_LENGTH	Length of the WITH OID clause of the typed view
OID_TEXT	WITH OID clause of the typed view
VIEW_TYPE_OWNER	Owner of the type of the view if the view is a typed view
VIEW_TYPE	Type of the view if the view is a typed view
SUPERVIEW_NAME	N/A

A.2.14 COLUMN_PRIVILEGES

Grants on columns for which the user is the grantor, grantee, or owner, or PUBLIC is the grantee:

column name	description
GRANTEE	Name of the user to whom access was granted
OWNER	Username of the object's owner
TABLE_NAME	Name of the object
COLUMN_NAME	Name of the column
GRANTOR	Name of the user who performed the grant
INSERT_PRIV	Permission to insert into the column
UPDATE_PRIV	Permission to update the column
REFERENCES_PRIV	Permission to reference the column
CREATED	Timestamp for the grant

A.2.15 DICTIONARY

List or data dictionary tables:

column name	description
TABLE_NAME	Table name
COMMENTS	Description of table

A.2.16 DUAL

column name	description
DUMMY	A dummy column

A.2.17 TABLE_PRIVILEGES

Grants on objects for which the user is the grantor, grantee, or owner, or PUBLIC is the grantee:

column name	description
GRANTEE	Name of the user to whom access is granted
OWNER	Owner of the object
TABLE_NAME	Name of the object
GRANTOR	Name of the user who performed the grant
SELECT_PRIV	Permission to select from an object
INSERT_PRIV	Permission to insert into an object
DELETE_PRIV	Permission to delete from an object
UPDATE_PRIV	Permission to update an object
REFERENCES_PRIV	N/A
ALTER_PRIV	Permission to alter an object
INDEX_PRIV	Permission to create or drop an index on an object
CREATED	Timestamp for the grant

A.2.18 USER_CATALOG

Tables, views, synonyms, and sequences owned by the use:

column name	description
TABLE_NAME	Name of the object
TABLE_TYPE	Type of object

A.2.19 USER_COL_COMMENTS

Comments on columns of user's tables and views:

column name	description
TABLE_NAME	Object name
COLUMN_NAME	Column name
COMMENTS	Comments on column

A.2.20 USER_CONSTRAINTS

Constraint definitions on user's tables:

column name	description
OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
CONSTRAINT_TYPE	Type of constraint definition
TABLE_NAME	Name associated with table with constraint definition
SEARCH_CONDITION	Text of search condition for table check
R_OWNER	Owner of table used in referential constraint
R_CONSTRAINT_NAME	Name of unique constraint definition for referenced table
DELETE_RULE	Delete rule for referential constraint
STATUS	Status of constraint
DEFERRABLE	Whether the constraint is deferrable
DEFERRED	Whether the constraint was initially deferred
VALIDATED	Whether all data obeys the constraint
GENERATED	Whether the name of the constraint is user or system generated
BAD	Constraint specifies a century in an ambiguous manner
LAST_CHANGE	When the constraint was last enabled or disabled
INDEX_OWNER	N/A
INDEX_NAME	N/A

A.2.21 USER_CONS_COLUMNS

Information about columns in constraint definitions owned by the user:

column name	description
OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
TABLE_NAME	Name associated with table with constraint definition
COLUMN_NAME	Name associated with column specified in the constraint definition
POSITION	Original position of column in definition

A.2.22 USER_INDEXES

Description of the user's own indexes:

column name	description
INDEX_NAME	Name of the index
INDEX_TYPE	Type of index
TABLE_OWNER	Owner of the indexed object
TABLE_NAME	Name of the indexed object
TABLE_TYPE	Type of the indexed object
UNIQUENESS	Uniqueness status of the index
COMPRESSION	N/A
PREFIX_LENGTH	0
TABLESPACE_NAME	Name of the tablespace containing the index
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
PCT_THRESHOLD	Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	Column ID of the last column to be included in index-organized table
FREELISTS	Number of process freelists allocated to this segment
FREELIST_GROUPS	Number of freelist groups allocated to this segment
PCT_FREE	N/A
LOGGING	Logging information
BLEVEL	Depth of the index from its root block to its leaf blocks. A depth of 1 indicates that the root and leaf block are the same.
LEAF_BLOCKS	Number of leaf blocks in the index
DISTINCT_KEYS	Number of distinct indexed values. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table.
AVG_LEAF_BLOCKS_PER	N/A
AVG_DATA_BLOCKS_PER	N/A
CLUSTERING_FACTOR	N/A
STATUS	State of the indexes: VALID
NUM_ROWS	Number of rows in the index
SAMPLE_SIZE	Size of the sample used to analyze the index
LAST_ANALYZED	Date on which this index was most recently analyzed

column name	description
DEGREE	Number of threads per instance for scanning the index
INSTANCES	Number of instances across which the index is to be scanned
PARTITIONED	Whether this index is partitioned
TEMPORARY	Whether the index is on a temporary table
GENERATED	Whether the name of the index is system generated
SECONDARY	N/A
BUFFER_POOL	Whether the index is a secondary object
USER_STATS	N/A
DURATION	N/A
PCT_DIRECT_ACCESS	N/A
ITYP_OWNER	N/A
ITYP_NAME	N/A
PARAMETERS	N/A
GLOBAL_STATS	N/A
DOMIDX_STATUS	N/A
DOMIDX_OPSTATUS	N/A
FUNCIDX_STATUS	N/A
JOIN_INDEX	N/A
IOT_REDUNDANT_PKEY_ELIM	N/A

A.2.23 USER_OBJECTS

Objects owned by the user:

column name	description
OBJECT_NAME	Name of object
SUBOBJECT_NAME	Name of the subobject
OBJECT_ID	Object number of the object
DATA_OBJECT_ID	Dictionary object number of the segment that contains the object
OBJECT_TYPE	Type of object
CREATED	N/A
LAST_DDL_TIME	N/A
TIMESTAMP	N/A
STATUS	State of the object: VALID
TEMPORARY	Whether the object is temporary
GENERATED	Was the name of this object system generated?
SECONDARY	N/A

A.2.24 USER_SYNONYMS

The user's private synonyms:

column name	description
SYNONYM_NAME	Name of the synonym
TABLE_OWNER	Owner of the object referenced by the synonym
TABLE_NAME	Name of the object referenced by the synonym
DB_LINK	N/A

A.2.25 USER_TABLES

Description of the user's own tables:

column name	description
TABLE_NAME	Name of the table
TABLESPACE_NAME	Name of the tablespace containing the table
CLUSTER_NAME	N/A
IOT_NAME	Name of the index organized table
PCT_FREE	N/A
PCT_USED	N/A
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
FREELISTS	Number of process freelists allocated to this segment
FREELIST_GROUPS	Number of freelist groups allocated to this segment
LOGGING	Logging information
BACKED_UP	N/A
NUM_ROWS	Number of rows in the table
BLOCKS	N/A
EMPTY_BLOCKS	N/A
AVG_SPACE	N/A
CHAIN_CNT	N/A
AVG_ROW_LEN	Average length of a row in the table in bytes
AVG_SPACE_FREELIST_BLOCKS	The average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	The number of blocks on the freelist
DEGREE	The number of threads per instance for scanning the table

column name	description
INSTANCES	The number of instances across which the table is to be scanned
CACHE	Whether the cluster is to be cached in the buffer cache
TABLE_LOCK	Whether table locking is enabled or disabled
SAMPLE_SIZE	Sample size used in analyzing this table
LAST_ANALYZED	Date on which this table was most recently analyzed
PARTITIONED	Indicates whether this table is partitioned
IOT_TYPE	If this is an index organized table
TEMPORARY	Can the current session only see data that it placed in this object itself?
SECONDARY	N/A
NESTED	If the table is a nested table
BUFFER_POOL	The default buffer pool for the object
ROW_MOVEMENT	N/A
GLOBAL_STATS	N/A
USER_STATS	N/A
DURATION	N/A
SKIP_CORRUPT	N/A
MONITORING	N/A
CLUSTER_OWNER	N/A
DEPENDENCIES	N/A
COMPRESSION	N/A

A.2.26 USER_TAB_COLUMNS

Columns of user's tables, views, and clusters:

column name	description
TABLE_NAME	Table, view, or cluster name
COLUMN_NAME	Column name
DATA_TYPE	Datatype of column
DATA_TYPE_MOD	Datatype modifier of the column
DATA_TYPE_OWNER	Owner of the datatype of the column
DATA_LENGTH	Maximum length of the column in bytes
DATA_PRECISION	N/A
DATA_SCALE	Digits to the right of decimal point in a number
NULLABLE	Does the column allow nulls? Value is <i>n</i> if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY key.
COLUMN_ID	Sequence number of the column as created
DEFAULT_LENGTH	N/A
DATA_DEFAULT	N/A
NUM_DISTINCT	Number of distinct values in each column of the table
LOW_VALUE	For tables with more than three rows, the second lowest and second highest values. These statistics are expressed in hexadecimal notation for the internal representation of the first 32 bytes of the values.
HIGH_VALUE	N/A
DENSITY	N/A
NUM_NULLS	The number of nulls in the column
NUM_BUCKETS	The number of buckets in histogram for the column
LAST_ANALYZED	The date on which this column was most recently analyzed
SAMPLE_SIZE	The sample size used in analyzing this column
CHARACTER_SET_NAME	The name of the character set
CHAR_COL_DECL_LENGTH	The length of the character set
GLOBAL_STATS	N/A
USER_STATS	N/A
AVG_COL_LEN	Average length of the column (in bytes)
CHAR_LENGTH	Displays the length of the column in characters
CHAR_USED	N/A

A.2.27 USER_TAB_COMMENTS

Comments on the tables and views owned by the user:

column name	description
TABLE_NAME	Name of the object
TABLE_TYPE	Type of object
COMMENTS	Comments on the object

A.2.28 USER_USERS

Information about the current user:

column name	description
USERNAME	Name of the user
USER_ID	N/A
ACCOUNT_STATUS	Indicates if the account is locked, expired or unlocked
LOCK_DATE	Date the account was locked
EXPIRE_DATE	Date of expiration of the account
DEFAULT_TABLESPACE	N/A
TEMPORARY_TABLESPACE	N/A
CREATED	N/A
EXTERNAL_NAME	User external name

A.2.29 USER_VIEWS

Text of views owned by the user:

column name	description
VIEW_NAME	Name of the view
TEXT_LENGTH	Length of the view text
TEXT	First line of the view text
TYPE_TEXT_LENGTH	Length of the type clause of the typed view
TYPE_TEXT	Type clause of the typed view
OID_TEXT_LENGTH	Length of the WITH OID clause of the typed view
OID_TEXT	WITH OID clause of the typed view
VIEW_TYPE_OWNER	Owner of the type of the view if the view is a typed view
VIEW_TYPE	Type of the view if the view is a typed view
SUPERVIEW_NAME	N/A

Sample Files

This appendix contains sample files of gateway initialization and Oracle Net **tnsnames.ora** and **listener.ora** file.

This appendix includes the following sections:

- [Sample Gateway Initialization File](#) on page B-2
- [Sample Oracle Net tnsnames.ora File](#) on page B-3
- [Sample Oracle Net listener.ora File](#) on page B-4

B.1 Sample Gateway Initialization File

The following sample Gateway Initialization File (initdrdaho1.ora) needs customization. For information on customizing this file, refer to ["Configuring the Host"](#) on page 12-4 in [Chapter 12, "Configuring the Gateway"](#). Also refer to [Appendix C](#).

```
#
# HS specific parameters
#
FDS_CLASS=TG4DRDA_DB2MVS
#TRACE_LEVEL=255
#LOG_DESTINATION=DB2.log
#ORACLE_DRDA_TCTL=debug.tctl
HS_COMMIT_POINT_STRENGTH=255
HS_NLS_DATE_FORMAT=YYYY-MM-DD
HS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P1
HS_RPC_FETCH_REBLOCKING=off
HS_RPC_FETCH_SIZE=32767
HS_FDS_FETCH_ROWS=20
#
# DRDA specific parameters
#
DRDA_CONNECT_PARM=DRDAON1
DRDA_REMOTE_DB_NAME=DB2V7R1
DRDA_PACKAGE_COLLID=ORACLE
DRDA_PACKAGE_NAME=G2DRSQL
DRDA_PACKAGE_CONSTOKEN=A92617CB3FE54701
DRDA_RECOVERY_USERID=ORADRDA
DRDA_RECOVERY_PASSWORD=ORADRDA
DRDA_ISOLATION_LEVEL=CS
#DRDA_PACKAGE_OWNER=ORADRDA
#DRDA_DISABLE_CALL=TRUE
```

B.2 Sample Oracle Net tnsnames.ora File

For information on tailoring your **tnsnames.ora** file for the gateway, refer to the instructions for ["Configuring Oracle Net"](#) on page 11-4.

```
ipc-ora9=(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=IPC)
    (KEY=ORAIPC)

  )
  (CONNECT_DATA=(SID=ORA101))
)
ipc-gtw=(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=IPC)
    (KEY=ORAIPC)

  )
  (CONNECT_DATA=(SID=drdahoal))
  (HS=)
)
```

B.3 Sample Oracle Net listener.ora File

For information on tailoring your **listener.ora** file for the gateway, refer to the instructions for ["Configuring Oracle Net"](#) on page 11-4.

```
#
# Sample listener.ora file for the Transparent Gateway for IBM DRDA
# Version Date: Jan-01-2002
# Filename: Listener.ora
#
LISTENER =
  (ADDRESS_LIST =
    (ADDRESS=
      (PROTOCOL= IPC)
      (KEY= ORAIPC))
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC=
      (SID_NAME=drdahoa1)
      (ORACLE_HOME=/oracle/tg4drda/10.1.0)
      (PROGRAM=g4drsrv)
    )
  )

STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF
```

This sample **listener.ora** file resides in the **\$ORACLE_HOME/network/admin** directory. If your listener uses the Oracle Net TCP/IP adapter instead of the IPC adapter, then replace the following lines under the LISTENER keyword:

```
    (ADDRESS=
      (PROTOCOL=IPC)
      (KEY=ORAIPC)
    )
with
    (ADDRESS=
      (PROTOCOL=TCP)
      (HOST=your_IP_node_name)
      (PORT=your_port_number)
    )
```

DRDA-Specific Parameters

This appendix contains the DRDA-specific parameters that are defined in the Gateway Initialization File. Read and understand the information on each parameter, taking special note of parameters that have defaults that do not apply to your system:

- [Modifying the Gateway Initialization File](#) on page C-2
- [Setting Parameters in the Gateway Initialization File](#) on page C-2
- [Syntax and Usage](#) on page C-2
- [Gateway Initialization File Parameters](#) on page C-2

C.1 Modifying the Gateway Initialization File

If you change any parameters in the Gateway Initialization File, then you must stop and re-start the Gateway in order for them to take effect. If you change certain parameters, then you must also rebind the DRDA package. Any parameters that affect the DRDA package have a note in their description that rebinding is required.

C.2 Setting Parameters in the Gateway Initialization File

Parameters that are specific to the gateway are stored in the Gateway Initialization File, **initsid.ora**.

C.3 Syntax and Usage

Parameters and their values are specified according to the syntax rules that are specified by Heterogeneous Services. The general form is:

```
[set] [private] drda_parameter = drda_parameter_value  
where:
```

`drda_parameter` is one of the DRDA parameters

`drda_parameter_value` is a character string with contents dependent upon the `drda_parameter`.

The **set** and **private** keywords are optional and have the following effect. If the **set** keyword is present, then the parameter and its value will be pushed into the process environment. This might be necessary if parameters from the Startup Shell Script are moved into the Gateway Initialization File. If the **private** keyword is present, then the parameter and its value will not be uploaded to the Oracle Database server. In general, Oracle corporation recommends that the **private** keyword not be used unless the parameter contains sensitive information (a userid or password, for example).

For further information on Heterogeneous Services and Initialization Parameters, refer to the section "Setting Initialization Parameters" in the *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

C.4 Gateway Initialization File Parameters

Following is a list of gateway-specific initialization file parameters and their descriptions. In addition to these parameters, generic Heterogeneous Services initialization file parameters may be set. Refer to the *Oracle Database Heterogeneous Connectivity Administrator's Guide* for a list of additional parameters.

C.4.1 DRDA_CACHE_TABLE_DESC

Default value: TRUE

Range of values: {TRUE | FALSE}

Syntax: DRDA_CACHE_TABLE_DESC= {TRUE | FALSE}

DRDA_CACHE_TABLE_DESC directs the gateway to cache table descriptions once per transaction. This can reduce the number of table lookups requested of the DRDA Server by Oracle and can speed up execution of SQL statements. You may wish to turn this option off if you will be altering the structure of a remote table and if you will be examining it within the same transaction.

C.4.2 DRDA_CAPABILITY

Default value: none

Range of values: Refer to ["Native Semantics"](#) on page 14-19

Syntax: DRDA_CAPABILITY={*FUNCTION*/{*ON*/*OFF*}},...

DRDA_CAPABILITY specifies which Oracle mapped functions will be treated natively. In other words, no special pre processing or post processing will be done for these functions. They will be passed through to the DRDA Server unmodified.

C.4.3 DRDA_CODEPAGE_MAP

Default value: codepage.map

Range of values: any valid file path

Syntax: DRDA_CODEPAGE_MAP=*codepage.map*

DRDA_CODEPAGE_MAP specifies the location of the codepage map. You may specify only the filename, which will be searched for within the \$ORACLE_HOME/tg4drda/admin directory, or you may specify the full path name of the file.

C.4.4 DRDA_COMM_BUFLLEN

Default value: 32767

Range of values: 512 through 32767

Syntax: DRDA_COMM_BUFLLEN=*num*

DRDA_COMM_BUFLLEN specifies the communications buffer length. This is a number indicating the size of the SNA send/receive buffer in bytes.

C.4.5 DRDA_CONNECT_PARM (SNA format)

Default value: DRDACON1

Range of values: any alphanumeric string 1 to 8 characters in length

Syntax: DRDA_CONNECT_PARM=*name*

DRDA_CONNECT_PARM specifies the Side Information name. Refer to [Chapter 6](#), [Chapter 7](#), [Chapter 8](#), and [Chapter 9](#) for details.

C.4.6 DRDA_CONNECT_PARM (TCP/IP format)

Default value: DRDACON1:446

Range of values: Any alphanumeric string 1 to 255 characters in length

Syntax: DRDA_CONNECT_PARM={*hostname*/*ip_address*}{:*port*}

DRDA_CONNECT_PARM specifies the TCP/IP hostname or IP Address of the DRDA Server and, as an option, the Service Port number on which the DRDA Server is listening.

C.4.7 DRDA_CMSRC_CM_IMMEDIATE

Default value: FALSE

Range of values: {TRUE | FALSE}

Syntax: DRDA_CMSRC_CM_IMMEDIATE={ *TRUE* / *FALSE*}

DRDA_CMSRC_CM_IMMEDIATE sets the SNA session allocation mode. A setting of FALSE will cause the gateway to wait for a free session if no free sessions exist. A setting of TRUE will cause the gateway to fail the allocation immediately if no free sessions exist.

C.4.8 DRDA_DEFAULT_CCSID

Default value: none

Range of values: any supported DRDA Server CCSID

Syntax: DRDA_DEFAULT_CCSID=*ccsid*

DRDA_DEFAULT_CCSID specifies the default CCSID or character set codepage for character set conversions when the DRDA Server database indicates that a character string has a CCSID of 65535. DRDA Servers use CCSID 65535 for columns specified as "FOR BIT DATA". In most cases, this parameter should not be specified, allowing CCSID 65535 to be treated as an Oracle RAW datatype.

This parameter is for supporting databases (in particular, DB2/400) that use CCSID 65535 as the default for all tables created. Allowing CCSID 65535 to be treated as another CCSID can save such sites from having to modify every table.

WARNING: Specifying any value for DRDA_DEFAULT_CCSID causes all "FOR BIT DATA" columns to be handled as text columns that need character set conversion and, therefore, any truly binary data in these columns can encounter conversion errors (ORA-28527).

C.4.9 DRDA_DESCRIBE_TABLE

Default value: TRUE

Range of values: {TRUE | FALSE}

Syntax: DRDA_DESCRIBE_TABLE={ *TRUE* / *FALSE*}

DRDA_DESCRIBE_TABLE directs the gateway to use the DRDA operation "Table Describe" to return the description of tables. This is an optimization that reduces the amount of time and resources that are used to look up the definition of a table.

Note: This feature is not compatible with DB2 Aliases or Synonyms. If you will be using DB2 aliases, then be sure to disable this option.

C.4.10 DRDA_DISABLE_CALL

Default value: TRUE

Range of values: {TRUE|FALSE}

Syntax: DRDA_DISABLE_CALL={ *TRUE* / *FALSE*}

DRDA_DISABLE_CALL controls stored procedure usage, and is also used to control how the package is bound on the target database. The gateway supports execution of stored procedures only on IBM DB2/OS390 Version 4.1 or later and DB2/400 Version 3.1 and later. Use the value, *FALSE*, only if the target database is DB2/OS390 Version 4.1 or later or DB2/400 Version 3.1 or later. Set this parameter to *TRUE* for all other target databases.

Rebinding Required: Any change to this parameter requires you to rebind.

C.4.11 DRDA_FLUSH_CACHE

Default value: SESSION

Range of values: {SESSION|COMMIT}

Syntax: DRDA_FLUSH_CACHE={ *SESSION* / *COMMIT*}

DRDA_FLUSH_CACHE specifies when the cursor cache is to be flushed. With DRDA_FLUSH_CACHE=COMMIT, the cursor cache is flushed whenever the transaction is committed. With DRDA_FLUSH_CACHE=SESSION, the cache is not flushed until the session terminates.

C.4.12 DRDA_GRAPHIC_PAD_SIZE

Default value: 0

Range of values: 0 through 127

Syntax: DRDA_GRAPHIC_PAD_SIZE=num

DRDA_GRAPHIC_PAD_SIZE is used to pad the size of a Graphic column as described by the DRDA Server. This is sometimes necessary depending upon the character set of the DRDA database and the Oracle database. If the Oracle database is based on EBCDIC and the DRDA database is based on ASCII, then a pad size of 2 may be needed.

C.4.13 DRDA_GRAPHIC_LIT_CHECK

Default value: FALSE

Range of values: {TRUE|FALSE}

Syntax: DRDA_GRAPHIC_LIT_CHECK={ TRUE|FALSE}

DRDA_GRAPHIC_LIT_CHECK directs the gateway to evaluate string literals within INSERT SQL statements in order to determine if they need to be converted to double-byte format for insertion into a Graphic column at the DRDA Server database. This is done by querying the column attributes of the table in the SQL statement to determine if a string literal is being applied to a column with a Graphic datatype. If the table column is Graphic, and if this parameter is TRUE, then the gateway will rewrite the SQL statement with the literal converted to double-byte format. Existing double-byte characters in the string will be preserved, and all single-byte characters will be converted to double-byte characters.

C.4.14 DRDA_GRAPHIC_TO_MBCS

Default value: FALSE

Range of values: {TRUE|FALSE}

Syntax: DRDA_GRAPHIC_TO_MBCS={ TRUE|FALSE}

DRDA_GRAPHIC_TO_MBCS directs the gateway to convert Graphic data that has been fetched from the DRDA Server into Oracle multi-byte data, translating double-byte characters into single-byte characters where possible.

C.4.15 DRDA_ISOLATION_LEVEL

Default value: CHG for DB2/400, CS for DB2/OS390, DB2/UDB, DB2/VM

Range of values: {CHG|CS|RR}

Syntax: DRDA_ISOLATION_LEVEL={ CHG|CS|RR}

DRDA_ISOLATION_LEVEL specifies the isolation level that is defined to the package when it is created. All SQL statements that are sent to the remote DRDA database are executed with this isolation level. Isolation level seriously affects performance of applications. Use caution when specifying an isolation level other than the default. For information on isolation levels, refer to your IBM database manuals.

The following table lists the isolation levels and their descriptions. The levels are specified in ascending order of control, with **CHG** having the least reliable cursor stability and **RR** having the most. Note that higher stability uses more resources on the server and can lock those resources for extended periods.

Table C–1 *Isolation Levels and Their Descriptions*

Level	Description
CHG	Change (default for DB2/400)
CS	Cursor Stability (default for DB2/UDB, DB2/OS390, and DB2/VM)
RR	Repeatable Read

Rebinding Required: Any change to this parameter requires you to rebind.

C.4.16 DRDA_LOCAL_NODE_NAME

Default value: AIX_RS6K

Range of values: any alphanumeric string 1 to 8 characters in length.

Syntax: DRDA_LOCAL_NODE_NAME=*name*

DRDA_LOCAL_NODE_NAME specifies the name by which the gateway will be known to the DRDA Server. This name is used internally by the DRDA Server to identify the local node.

C.4.17 DRDA_MBCS_TO_GRAPHIC

Default value: FALSE

Range of values: {TRUE | FALSE}

Syntax: DRDA_MBCS_TO_GRAPHIC={ *TRUE* / *FALSE* }

DRDA_MBCS_TO_GRAPHIC directs the gateway to convert multi-byte data (that has been sent from Oracle to the DRDA database) into pure double-byte characters. This parameter is primarily intended to be used with bind variables in order to ensure that the data is properly formatted and will therefore be acceptable to the DRDA Server. It applies only to INSERT SQL statements that are using bind variables. When used in combination with the DRDA_GRAPHIC_LIT_CHECK parameter, this parameter can help ensure that data that is being inserted into a Graphic column is handled correctly by the target DRDA Server.

C.4.18 DRDA_OPTIMIZE_QUERY

Default value: TRUE

Range of values: {TRUE | FALSE}

Syntax: DRDA_OPTIMIZE_QUERY={ *TRUE* / *FALSE* }

DRDA_OPTIMIZE_QUERY turns on or off the distributed query optimizer (DQO) capability. Refer to ["Performing Distributed Queries"](#) on page 13-4 in [Chapter 13, "Using the Gateway"](#). The DQO capability is useful for optimizing queries that access large amounts of data, but it can add overhead to small queries.

This parameter is valid only if your DRDA Server is DB2/OS390 or DB2/VM. If your DRDA Server is DB2/400 or DB2/UDB, then you must set the value to `FALSE`.

C.4.19 DRDA_PACKAGE_COLLID

Default value: ORACLE

Range of values: an alphanumeric string 1 to 18 characters in length

Syntax: DRDA_PACKAGE_COLLID=*collection_id*

DRDA_PACKAGE_COLLID specifies the package collection ID. Note that in DB2/400, the collection ID is actually the name of an AS/400 library.

Rebinding Required: Any change to this parameter requires you to rebind the package.

C.4.20 DRDA_PACKAGE_CONSTOKEN

Default value: none, use the sample provided

Range of values: a 16-digit hexadecimal number

Syntax: DRDA_PACKAGE_CONSTOKEN=*hexnum*

DRDA_PACKAGE_CONSTOKEN specifies the package consistency token. This is a 16-digit hexadecimal representation of an 8-byte token. Oracle Corporation recommends that you do not change the consistency token. The consistency token used at runtime must match the one used when the package is bound. The value depends on the DRDA Server being used.

Rebinding Required: Any change to this parameter requires you to rebind the package.

C.4.21 DRDA_PACKAGE_NAME

Default value: G2DRSQL

Range of values: an alphanumeric string 1 to 18 characters in length

Syntax: DRDA_PACKAGE_NAME=*name*

DRDA_PACKAGE_NAME specifies the package name. Note that the package is stored in the DRDA Server under this name as a SQL resource. Refer to the DRDA Server documentation for length limitations on package names. Many typical implementations restrict the length to 8 characters.

Rebinding Required: Any change to this parameter requires that you rebind the package.

C.4.22 DRDA_PACKAGE_OWNER

Default value: none

Range of values: any valid user ID

Syntax: DRDA_PACKAGE_OWNER=*userid*

DRDA_PACKAGE_OWNER specifies the database user ID that owns the package. This allows the owner to be a user other than the connected user ID when the package is created. The package owner must be the same user as the owner of the ORACLE2PC table. This is not valid for DB2/VM.

Rebinding Required: Any change to this parameter requires you to rebind the package.

C.4.23 DRDA_PACKAGE_SECTIONS

Default value: 100

Range of values: any integer between 1 and 65535

Syntax: DRDA_PACKAGE_SECTIONS=*num*

DRDA_PACKAGE_SECTIONS specifies the number of cursors declared at the remote database when the package is bound. This is the maximum number of open cursors allowed at any one time. Change this parameter only if an application needs more than 100 open concurrent cursors.

Rebinding Required: Any change to this parameter requires you to rebind the package.

C.4.24 DRDA_READ_ONLY

Default value: FALSE

Range of values: {TRUE | FALSE}

Syntax: DRDA_READ_ONLY= { *TRUE* / *FALSE* }

DRDA_READ_ONLY specifies whether the gateway runs in a read-only transaction mode. In this mode, SQL statements which modify data are not allowed.

C.4.25 DRDA_RECOVERY_PASSWORD

Default value: none

Range of values: any valid password

Syntax: DRDA_RECOVERY_PASSWORD=*passwd*

DRDA_RECOVERY_PASSWORD is used with the DRDA_RECOVERY_USERID. The recovery user connects to the IBM database if a distributed transaction is in doubt. For more information, refer to ["Two-Phase Commit Processing"](#) on page 13-5. Also refer to [Chapter 15](#) for information about security and about encrypting passwords.

C.4.26 DRDA_RECOVERY_USERID

Default value: ORARECOV

Range of values: any valid user ID

Syntax: DRDA_RECOVERY_USERID=*userid*

DRDA_RECOVERY_USERID specifies the user ID that is used by the gateway if a distributed transaction becomes in doubt. This user ID must have execute privileges on the package and must be defined to the IBM database.

If a distributed transaction becomes in doubt, then the Oracle integrating server determines the status of the transaction by connecting to the IBM database, using the DRDA_RECOVERY_USERID. If this parameter is missing, the gateway attempts to connect to a user ID of ORARECOV. For more information, refer to ["Two-Phase Commit Processing"](#) on page 13-5.

C.4.27 DRDA_REMOTE_DB_NAME

Default value: DB2V2R3

Range of values: an alphanumeric string 1 to 18 characters in length

Syntax: DRDA_REMOTE_DB_NAME=*name*

DRDA_REMOTE_DB_NAME specifies the DRDA Server location name. This is an identifying name that is assigned to the server for DRDA purposes. A technique for determining this name by using a SQL SELECT statement is discussed in each of the server-specific installation sections in [Chapter 5, "Configuring the DRDA Server"](#).

C.4.28 DRDA_SECURITY_TYPE

Default value: PROGRAM

Range of values: {PROGRAM|SAME}

Syntax: DRDA_SECURITY_TYPE={ *PROGRAM* / *SAME* }

DRDA_SECURITY_TYPE specifies the type of security used for SNA communications. For more information about types of security and about setting DRDA_SECURITY_TYPE, refer to [Chapter 15, "Security Considerations"](#). Also refer to Chapter 15 for information on security and encrypting passwords.

C.4.29 FDS_CLASS

Default value: TG4DRDA_DB2MVS

Range of values: Refer to the list below for valid values

Syntax: FDS_CLASS=*TG4DRDA_DB2MVS*

FDS_CLASS specifies the capability classification used by the Oracle Database server and the gateway. These values might change from release to release, depending upon whether the gateway capabilities change.

The valid default values for FDS_CLASS are as follows:

For a DB2/OS390 database: TG4DRDA_DB2MVS

For a DB2/VM database: TG4DRDA_DB2VM

For a DB2/400 database: TG4DRDA_DB2400

For a DB2/UDB database: TG4DRDA_DB2UDB

C.4.30 FDS_CLASS_VERSION

Default value: 10.1.0.2.0

Range of values: 10.1.0.2.0

Syntax: FDS_CLASS_VERSION=1

FDS_CLASS_VERSION specifies the version of the FDS_CLASS capabilities. Do not specify this parameter unless directed to do so by Oracle Support Services.

C.4.31 FDS_INSTANCE

Default value: DRD1

Range of values: the name of the gateway SID

Syntax: FDS_INSTANCE=*drdahoa1*

FDS_INSTANCE specifies a subset of the FDS_CLASS capabilities that may be modified by the user, based on initialization file parameters. If you do not specify this parameter, then its value will be the Oracle SID that is defined in the TNS Listener entry.

C.4.32 HS_FDS_FETCH_ROWS

Default value: 20

Range of values: any integer between 1 and 1000

Syntax: HS_FDS_FETCH_ROWS=*num*

HS_FDS_FETCH_ROWS specifies the fetch array size. This is the number of rows to fetch at one time from the DRDA Server and to return to the Oracle Database server. This parameter will be affected by the HS_RPC_FETCH_SIZE and HS_RPC_FETCH_REBLOCKING parameters. For further information on these parameters, refer to the section "Controlling the Array Fetch Between Agent and Non-Oracle Database server" in the *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

C.4.33 HS_LANGUAGE

Default value: none

Range of values: any valid language specification

Syntax: HS_LANGUAGE=*language[_territory.character_set]*

HS_LANGUAGE specifies the language and the character set that the gateway will use to interact with the DRDA Server. Care must be taken in choosing the value of these parameters, especially when the gateway will be accessing GRAPHIC data. For additional details, refer to [Appendix D](#) and to the *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

C.4.34 HS-NLS_NCHAR

Default value: none

Range of values: any valid character set specification

Syntax: HS-NLS_NCHAR=*character_set*

HS-NLS_NCHAR specifies the character set that the gateway will use to interact with the DRDA Server when accessing GRAPHIC data. Set this parameter to the same value as the character set component of the HS_LANGUAGE parameter. For additional details, refer to [Appendix D](#) and to the *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

C.4.35 LOG_DESTINATION

Default value: \$ORACLE_HOME/tg4drda/log/gateway sid_pid.log

Range of values: any valid file path

Syntax: LOG_DESTINATION=*logpath*

LOG_DESTINATION specifies the destination for gateway logging and tracing. This parameter should specify a file. If the file already exists, it will be overwritten.

After any failure to open the logpath, a second attempt to open the default is made.

Usually, LOG_DESTINATION should specify a directory. If it is specified as a file, and if two or more users simultaneously use the same instance of the gateway, then they are writing to the same log. The integrity of this log is not guaranteed. If you do not specify this parameter, then the default is assumed.

C.4.36 ORA_MAX_DATE

Default value: 4712-12-31

Range of values: any valid date less than 4712-12-31

Syntax: ORA_MAX_DATE=*yyyy-mm-dd*

ORA_MAX_DATE specifies the gateway maximum date value. If the fetched date value is larger than 4712-12-31, the gateway replaces the date value with the value defined by the ORA_MAX_DATE parameter. Any date between January 1, 4712 BC and December 31, 4712 AD is valid.

C.4.37 ORA_NLS33

Default value: \$ORACLE_HOME/nls/data

Range of values: any valid NLS directory path

Syntax: SET ORA_NLS33=*nlspace*

ORA_NLS33 specifies the directory to which the gateway loads its character sets and other language data. Normally this parameter does not need to be set. Some configurations, however, may require that it be set.

C.4.38 ORACLE_DRDA_TCTL

Default value: none

Range of values: any valid file path

Syntax: ORACLE_DRDA_TCTL=*tracecontrolpath*

ORACLE_DRDA_TCTL specifies the path to the DRDA internal trace control file. This file contains module tracing commands. A sample file is stored in \$ORACLE_HOME/tg4drda/admin/debug.tctl. This parameter is used for diagnostic purposes.

C.4.39 ORACLE_DRDA_TRACE

Default value: value specified for LOG_DESTINATION

Range of values: any valid file path

Syntax: ORACLE_DRDA_TRACE=*logpath*

ORACLE_DRDA_TRACE is used to specify a different log path for DRDA internal tracing. This tracing is separate from the rest of the gateway tracing, as specified by the LOG_DESTINATION parameter. By default, this parameter will append the DRDA internal trace to the gateway trace. This parameter is used for diagnostic purposes.

C.4.40 TRACE_LEVEL

Default Value: 0

Range of values: 0-255

Syntax: TRACE_LEVEL=*number*

TRACE_LEVEL specifies a code tracing level. This value determines the level of detail which is logged to the gateway logfile during execution. This parameter is primarily used for diagnostics.

National Language Support

This appendix documents the National Language Support (NLS) information for the Oracle Transparent Gateway for DRDA. This supplements the general Oracle NLS information found in the *Oracle Database Application Developer's Guide - Fundamentals*.

National Language Support enables Oracle applications to interact with users in their native language, using their conventions for displaying data. The Oracle NLS architecture is data-driven, enabling support for specific languages and character encoding schemes to be added without any changes in source code.

There are a number of different settings in the gateway, DRDA Server, Oracle Database 10g server, and client that affect NLS processing. In order for translations to take place correctly, character settings of these components must be compatible.

This appendix contains the following sections:

- [Overview of NLS Interactions](#) on page D-2
- [Client and Oracle Integrating Server Configuration](#) on page D-4
- [Gateway Language Interaction with DRDA Server](#) on page D-4
- [Gateway Codepage Map Facility](#) on page D-7
- [Multi-Byte and Double-Byte Support in the Gateway](#) on page D-10
- [Message Availability](#) on page D-12
- [Example of NLS Configuration](#) on page D-12

D.1 Overview of NLS Interactions

Figure D–1 illustrates NLS interactions within your system, including each component of your system and the parameters of each component that affect NLS processing in a distributed environment. Table D–1 describes the architecture illustrated in Figure D–1.

Figure D–1 Architecture of NLS Interactions with Your System Components

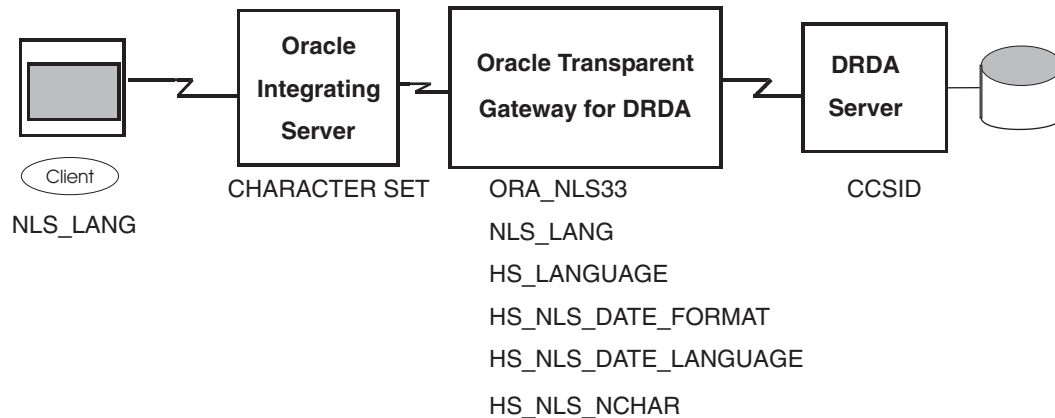


Table D–1 describes in detail the parameters and variables needed for NLS processing within each of your system's environments: the client environment, the Oracle integrating server, the gateway and the DRDA Server.

Parameters Needed for NLS Processing in Your System's Environments

Table D–1 Parameters Needed for NLS Processing in Your System's Environments

Environment	Parameter or Variable	Description
Client	NLS_LANG	An environmental variable. NLS_LANG sets the NLS environment used by the database both for the server session and for the client application. This ensures that the language environments of both database and client application are automatically the same. Because NLS_LANG is an environment variable, it is read by the client applications at startup time. The client communicates the information defined in NLS_LANG to the server when it connects. For detailed information, refer to "Client and Oracle Integrating Server Configuration" on page D-4.
Oracle integrating server	CHARACTER SET	This option is set during creation of the database. CHARACTER SET determines the character encoding scheme used by the database and is defined at database creation in the CREATE DATABASE statement. All data columns of type CHAR, VARCHAR2, and LONG have their data stored in the database character set. For detailed information, refer to "Client and Oracle Integrating Server Configuration" on page D-4.
Oracle Transparent Gateway for DRDA	ORA_NLS33	An environmental variable. ORA_NLS33 determines where the gateway loads its character sets and other language data. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page D-4.

Table D-1 (Cont.) Parameters Needed for NLS Processing in Your System's Environments

Environment	Parameter or Variable	Description
Oracle Transparent Gateway for DRDA	NLS_LANG	An environmental variable. NLS_LANG defines the character set used for communication between the gateway and the Oracle integrating server. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page D-4.
Oracle Transparent Gateway for DRDA	HS_LANGUAGE	An initialization parameter. HS_LANGUAGE defines the character set used for communication between the gateway and the DRDA Server. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page D-4.
Oracle Transparent Gateway for DRDA	HS_NLS_NCHAR	An initialization parameter. HS_NLS_NCHAR defines the NCHAR character set that is used for communications between the gateway and the DRDA Server. This parameter is required when the gateway will be accessing GRAPHIC or multi-byte data on the DRDA Server. Set this parameter to the same value as the character set component of the HS_LANGUAGE parameter. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page D-4.
Oracle Transparent Gateway for DRDA	HS_NLS_DATE_FORMAT	An initialization parameter. HS_NLS_DATE_FORMAT specifies the format for dates used by the DRDA Server. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page D-4.
Oracle Transparent Gateway for DRDA	HS_NLS_DATE_LANGUAGE	An initialization parameter. HS_NLS_DATE_LANGUAGE specifies the language used by the DRDA Server for day and month names, and for date abbreviations. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page D-4.
DRDA Server	CCSID	CCSID is the server character set that is mapped in the gateway to the equivalent Oracle character set. The CCSID specifies the character set that the DRDA database uses to store data. It is defined when you create your database. For detailed information, refer to "Gateway Codepage Map Facility" on page D-7.

D.2 Client and Oracle Integrating Server Configuration

A number of NLS parameters control NLS processing between the Oracle Database server and client. You can set language-dependent behavior defaults for the server, and you can set language-dependent behavior for the client that overrides these defaults. For a complete description of NLS parameters, refer to the NLS chapter in the *Oracle Database Administrator's Guide*. These parameters do not directly affect gateway processing. However, you must ensure that the client character set (which is specified by the Oracle Database server NLS_LANG environment variable) is compatible with the character sets that you specify on the gateway and on the DRDA Server.

When you create your Oracle Database, the character set that is used to store data is specified by the CHARACTER SET clause of the CREATE DATABASE statement. After the database is created, the database character set cannot be changed unless you re-create the database.

Normally, the default for CHARACTER SET is US7ASCII, which supports only the 26 Latin alphabetic characters. If you have specified 8-bit character sets on the gateway and the DRDA Server, then you must have a compatible 8-bit character set defined on your database. To check the character set of an existing database, issue the command:

```
SELECT USERENV('LANGUAGE') FROM DUAL;
```

For more information, refer to "Specifying Character Sets" in the *Oracle Database Administrator's Guide*.

Note that this does not mean that the gateway character set must be the same as the Oracle Database server character set. The Oracle Net facility will be performing implicit conversion between the Oracle Database server character set and the gateway character set.

D.3 Gateway Language Interaction with DRDA Server

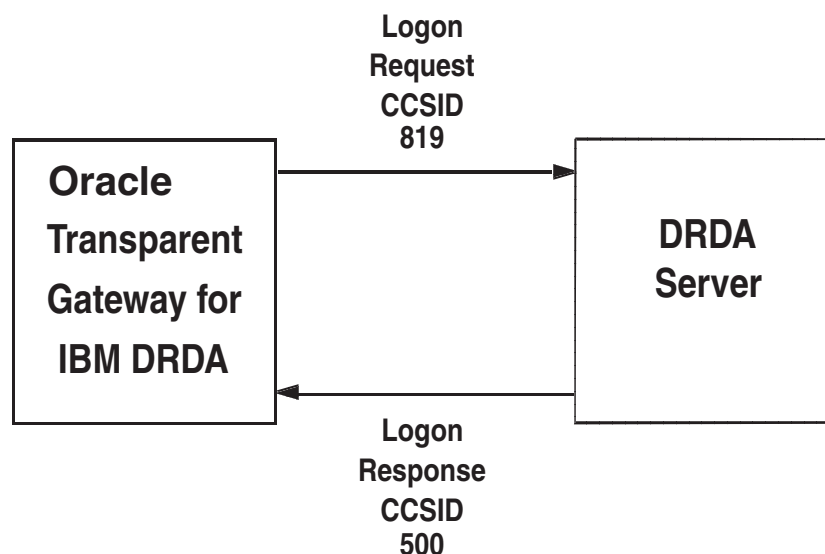
During logon of the gateway to the DRDA Server, initial language information is exchanged between the Gateway and the server. First, the Gateway sends to the DRDA Server the CCSID in which it will be conversing. In the following example, the Oracle Character Set "WE8ISO8859P1" is mapped to CCSID 819 (an ASCII Code Page). This CCSID is sent to the DRDA Server. The DRDA Server responds with the CCSID in which it will be conversing. This will be the CCSID with which the DB2 database was generated. Also in the following example, this is CCSID 500, an EBCDIC Code Page. [Figure D-2, "Gateway Language Interaction with DRDA Server"](#), illustrates this process.

A DB2 instance will map unknown CCSIDs using the SYSIBM.SYSSTRINGS table (This table has different names for the various DB2 versions). It is possible to add additional character set mappings to this table by using DB2 utilities. Please refer to the DB2 Installation documentation for details.

The setting of the HS_LANGUAGE parameter in the gateway **initsid.ora** determines which CCSID is used by the Gateway for the conversation. Similarly, the setting of the HS_NLS_NCHAR parameter determines which CCSID will be used by the gateway for GRAPHIC data interchange. For the list of supported ASCII-based Oracle Character Sets that are mapped to CCSIDs, refer to ["Gateway Codepage Map Facility"](#) on page D-7.

Note again that it is not necessary for the gateway character set to be the same as the Oracle Database server character set. In many cases, it is not feasible to set the gateway character set equal to the Oracle Database server character set because the DRDA Server will not have a valid translation for it. Instead, choose a character set that will have the most complete intersection with the character set that is used by the DRDA Server. The Oracle Net facility will do any translation between the gateway character set and the Oracle Database server character set.

Figure D-2 Gateway Language Interaction with DRDA Server



D.3.1 Gateway Configuration

After the gateway is installed, there are several parameters that you must change in order to customize for NLS support.

D.3.2 NLS Parameters in the Gateway Initialization File

There are three parameters in the Gateway Initialization File, *initsid.ora*, that affect NLS:

- HS_LANGUAGE
- HS-NLS_NCHAR
- HS-NLS_DATE_FORMAT
- HS-NLS_DATE_LANGUAGE

D.3.2.1 HS_LANGUAGE

HS_LANGUAGE defines the character set that is used for communication between the gateway and the DRDA Server. It specifies the conventions such as: the language used for messages from the target system; names of days and months; symbols for AD, BC, AM, and PM; and default language sorting mechanism.

The syntax of the HS_LANGUAGE parameter is:

```
HS_LANGUAGE=language[_territory.character_set]
```

where:

language can be any valid language.

territory is optional, and defaults to AMERICA.

character_set is optional and defaults to US7ASCII. This must be an ASCII base character set name, and it should match a character set listed in the gateway codepage map. Refer to "[Gateway Codepage Map Facility](#)" on page D-7 for the list of supplied character set mappings.

If you omit the HS_LANGUAGE parameter from **initsid.ora**, then the default setting is AMERICAN_AMERICA.US7ASCII. EBCDIC character sets are not supported. The values for *language* and *territory* (such as AMERICAN_AMERICA) must be valid, but they have no effect on translations.

D.3.2.2 HS_NLS_NCHAR

HS_NLS_NCHAR specifies the character set that is used by the gateway to interchange GRAPHIC data. For correct compatibility, set it to the same character set name that is specified in the HS_LANGUAGE parameter. If it is set to a character set other than that specified in HS_LANGUAGE, or if it is omitted, then translation errors will occur.

D.3.2.3 HS_NLS_DATE_FORMAT

HS_NLS_DATE_FORMAT specifies the format for dates used by the DRDA Server.

The syntax of the NLS_DATE_FORMAT parameter is:

```
HS_NLS_DATE_FORMAT=date_format
```

where *date_format* must be YYYY-MM-DD, the ISO date format. If this parameter is set to any other value or is omitted, then you receive an error when updating, deleting from, selecting from, or inserting into, a table with date columns.

D.3.2.4 HS_NLS_DATE_LANGUAGE

HS_NLS_DATE_LANGUAGE specifies the language used by the DRDA Server for day and month names, and for date abbreviations. Because ISO date format contains numbers only, this parameter has no effect on gateway date processing and should be omitted.

D.4 Gateway Codepage Map Facility

The gateway now has a user specifiable facility to map IBM Coded Character Set Identifiers (CCSIDs) to Oracle Character Sets for the purpose of data translation.

The map name defaults to "codepage.map" and is located in the directory `$ORACLE_HOME/tg4drda/admin`. Refer to [Appendix C, "DRDA-Specific Parameters"](#) for more detailed information about the `DRDA_CODEPAGE_MAP` parameter.

The map has two different forms of syntax. The first form of syntax defines a mapping between a CCSID and an Oracle Database character set:

```
[S|D|M] CCSID direction Oracle_CharacterSet {shift}
```

where:

S designates a single-byte character set

D designates a double-byte character set

M designates a multi-byte character set

CCSID is the IBM coded character set identifier

direction is one of the following:

- = means mapping is bidirectional
- < means mapping is one-way, Oracle character set to CCSID
- > means mapping is one-way, CCSID to Oracle character set

Oracle_CharacterSet is the name of a valid Oracle Character Set.

shift indicates a character set that requires Shift OUT/IN processing. Set this attribute only for EBCDIC-based double-byte and multi-byte mappings.

The second form of syntax defines a mapping of a multi-byte CCSID to its single-byte and double-byte CCSID equivalents:

```
MBC multi = single double
```

where:

multi is the multi-byte CCSID

single is the single-byte CCSID

double is the double-byte CCSID

This facility is intended as a way of mapping CCSIDs which were not previously mapped as shipped with the gateway. You must contact Oracle Support Services before modifying this map.

The following are the contents of the map as shipped with the Oracle Transparent Gateway for DRDA;

```
# Copyright (c) 2001, 2003, Oracle Corporation. All rights reserved.
# Transparent Gateway for IBM DRDA - CodePage/Oracle CharacterSet Map
# S==Single-byte, D==Double-byte, M==Multi-byte, MBC==SBC DBC mapping
#
# Single-byte codepage mappings
#
S 37 = WE8EBCDIC37 # United States/Canada EBCDIC
S 273 = D8EBCDIC273 # Austria/Germany EBCDIC
S 277 = DK8EBCDIC277 # Denmark/Norway EBCDIC
S 278 = S8EBCDIC278 # Finland/Sweden EBCDIC
S 280 = I8EBCDIC280 # Italy EBCDIC
S 284 = WE8EBCDIC284 # Latin America/Spain EBCDIC
S 285 = WE8EBCDIC285 # United Kingdom EBCDIC
S 297 = F8EBCDIC297 # France EBCDIC
#S 420 = AR8EBCDICX # Arabic Bilingual (USA English) EBCDIC
S 420 = AR8XBASIC # Arabic Bilingual (USA English) EBCDIC
S 424 = IW8EBCDIC424 # Israel (Hebrew) EBCDIC
S 437 = US8PC437 # Personal Computer,USA ASCII
S 500 = WE8EBCDIC500 # International EBCDIC
S 813 = EL8ISO8859P7 # Greek ASCII
S 819 = WE8ISO8859P1 # ISO/ANSI Multilingual ASCII
S 838 = TH8TISEBCDIC # Thai w/Low-Tone Marks & Ancient Chars EBCDIC
S 850 < US7ASCII # Multilingual Page - Personal Computer ASCII
S 850 = WE8PC850 # Multilingual Page - Personal Computer ASCII
S 864 = AR8ISO8859P6 # Arabic - Personal Computer ASCII
S 870 = EE8EBCDIC870 # Latin 2, Multilingual/ROECE EBCDIC
S 871 = WE8EBCDIC871 # Iceland - CECP EBCDIC
S 875 = EL8EBCDIC875 # Greece EBCDIC
S 904 > US7ASCII # Traditional Chinese - PC-Data ASCII
S 912 = EE8ISO8859P2 # Latin 2 8-bit ASCII
S 916 = IW8ISO8859P8 # Israel (Hebrew) ASCII
S 1025 = CL8EBCDIC1025 # Cyrillic, Multiling EBCDIC
S 1086 = IW8EBCDIC1086 # Israel EBCDIC
S 1252 = WE8MSWIN1252 # Latin 1 - MS-Windows ASCII
S 1253 = EL8MSWIN1253 # Greek - MS-Windows ASCII
S 28709 > WE8EBCDIC37 # United States/Canada (CP28709==CP37) EBCDIC
#
# Multi-byte codepage mappings
#
#S 833 > K016DBCS # Korean Extended single-byte EBCDIC
#D 834 > K016DBCS shift # Korean double-byte EBCDIC
#M 933 = K016DBCS shift # Korean Mixed multi-byte EBCDIC

#MBC 933 = 833 834 # Korean Mixed multi-byte EBCDIC
#
#S 1088 > K016MSWIN949 # Korean KS single-byte PC-Data ASCII
#D 951 > K016MSWIN949 # Korean KS double-byte PC-Data ASCII
#M 949 = K016MSWIN949 # Korean KS multi-byte PC-Data ASCII
#MBC 949 = 1088 951 # Korean KS multi-byte PC-Data ASCII
#
#S 891 > K016KSC5601 # Korean single-byte ASCII
#S 1040 > K016KSC5601 # Korean single-byte ASCII
#D 926 > K016KSC5601 # Korean double-byte ASCII
#M 934 = K016KSC5601 # Korean multi-byte ASCII
#M 944 > K016KSC5601 # Korean multi-byte ASCII
#MBC 934 = 891 926 # Korean multi-byte ASCII
#MBC 944 = 1040 926 # Korean multi-byte Extended ASCII
#
```

```

#S 28709 > ZHT16DBCS      # Traditional Chinese single-byte      EBCDIC
#D  835 > ZHT16DBCS shift # Traditional Chinese double-byte      EBCDIC
#M  937 = ZHT16DBCS shift # Traditional Chinese multi-byte        EBCDIC
#MBC 937 = 28709 835      # Traditional Chinese multi-byte        EBCDIC
#
#S 1114 > ZHT16MSWIN950 # Traditional Chinese single-byte      ASCII
#D  947 > ZHT16MSWIN950 # Traditional Chinese double-byte      ASCII
#M  950 = ZHT16MSWIN950 # Traditional Chinese multi-byte        ASCII
#MBC 950 = 1114 947      # Traditional Chinese multi-byte        ASCII
#
#S  836 > ZHS16DBCS      # Simplified Chinese single-byte      EBCDIC
#D  837 > ZHS16DBCS shift # Simplified Chinese double-byte      EBCDIC
#M  935 = ZHS16DBCS shift # Simplified Chinese multi-byte        EBCDIC
#MBC 935 = 836 837      # Simplified Chinese multi-byte        EBCDIC
#
#S 1027 > JA16DBCS      # Japanese single-byte          EBCDIC
#D  300 > JA16DBCS shift # Japanese double-byte          EBCDIC
#D 4396 > JA16DBCS shift # Japanese double-byte          EBCDIC
#M  939 = JA16DBCS shift # Japanese multi-byte          EBCDIC
#M 5035 > JA16DBCS shift # Japanese multi-byte          EBCDIC
#MBC 939 = 1027 300      # Japanese multi-byte          EBCDIC
#MBC 5035 = 1027 4396    # Japanese multi-byte          EBCDIC
#
#S  290 > JA16EBCDIC930  # Japanese single-byte      EBCDIC
#D  300 > JA16EBCDIC930 shift # Japanese double-byte      EBCDIC
#D 4396 > JA16EBCDIC930 shift # Japanese double-byte      EBCDIC
#M  930 = JA16EBCDIC930 shift # Japanese multi-byte      EBCDIC
#M 5026 > JA16EBCDIC930 shift # Japanese multi-byte      EBCDIC
#MBC 930 = 290 300        # Japanese multi-byte      EBCDIC
#MBC 5026 = 290 4396      # Japanese multi-byte      EBCDIC
#

```

Refer to the following list to check the character set of an existing database:

- **for DB2/OS390:** Ask your system administrator. There is no single command you use.
- **for DB2/400:** Issue the command `DSPSYSVAL SYSVAL (QCCSID)`
- **for DB2/UDB:** Ask your system administrator. There is no single command you use.
- **for DB2/VM:** Issue the statement `ID`. This shows you the default CCSIDs used at startup.

D.5 Multi-Byte and Double-Byte Support in the Gateway

In order to enable the gateway to properly handle double-byte and multi-byte data, you must configure the codepage map facility with proper multi-byte maps and (as an option) you can set the following gateway configuration parameters:

- DRDA_GRAPHIC_LIT_CHECK
- DRDA_GRAPHIC_TO_MBCS
- DRDA_MBCS_TO_GRAPHIC
- DRDA_GRAPHIC_PAD_SIZE

Refer to [Appendix C, "DRDA-Specific Parameters"](#), for the values of these parameters.

Configuring the codepage map requires knowledge of the codepages that have been configured in the DRDA Server database as well as knowledge of compatible Oracle Database Character sets.

IBM coded character set identifiers (CCSIDs) are used to indicate which codepages are configured as the primary codepage for the database, as well as any translation character sets loaded into the database. Some DRDA Servers, such as with DB2, have a translation facility in which character set transforms are mapped between two compatible character sets. For DB2/OS390, these transforms are stored in the table SYSIBM.SYSSTRINGS and transform on the CCSID codepage to another CCSID codepage. In SYSSTRINGS, **IN** and **OUT** columns specify the CCSIDs that are used in the transform. Typical transforms are from ASCII to EBCDIC and back again. Two transforms are therefore used for two given CCSIDs.

Multi-byte codepages are a composite of a single-byte codepage and a double-byte codepage. As an example, the Korean EBCDIC multi-byte codepage, CCSID 933, is composed of two codepages: codepage 833 (for single-byte) and codepage 834 (for double-byte). The DRDA Server, therefore, can send data to the gateway in any of these three codepages, and the gateway must translate appropriately depending upon which codepage the data is associated with. Because CCSID 933 is an EBCDIC-based codepage, and the gateway must use an ASCII-based codepage, we identify an equivalent set of codepages, which are ASCII-based. An example would be the Korean multi-byte codepage, CCSID 949, which is composed of two codepages: codepage 1088 (for single-byte) and codepage 951 (for double-byte).

The codepage map facility is used to map these CCSIDs into the equivalent Oracle Database Character Sets. Unlike IBM CCSIDs, Oracle Database Character Sets are unified (in that single-byte and double-byte character sets have been combined into one set) and are thus identified by one ID instead of three IDs. In our previous example, the equivalent Oracle Database Character Set for the ASCII Korean codepages would be KO16MSWIN949, and the EBCDIC Korean codepages would be KO16DBCS. These are identified to the gateway by using a set of mapping entries in the **codepage.map** file.

First, the EBCDIC Korean sets are:

```
S  833 > KO16DBCS          # Korean Extended single-byte      EBCDIC
D  834 > KO16DBCS shift    # Korean double-byte              EBCDIC
M  933 = KO16DBCS shift    # Korean Mixed multi-byte         EBCDIC
MBC 933 = 833 834          # Korean Mixed multi-byte         EBCDIC
```

Notice that the multi-byte set is a bidirectional map to KO16DBCS, while the single and double codepages are mapped one-way to KO16DBCS. Because only one bidirectional CCSID to Oracle Database Character Set entry for a given pair can exist, we directly map the multi-byte sets. And because the single-byte and double-byte

CCSIDs are ostensibly subsets of KO16DBCS, we map them as one-way entries. Note that double-byte and multi-byte maps are tagged with the "shift" attribute. This is required for EBCDIC double-byte and multi-byte codepages as part of the shift out/in encapsulation of data. Note that the single-byte map is not marked because single-byte sets are not allowed to contain double-byte data and thus will never use shift encapsulation. Also note that the MBC entry ties the codepages together.

The ASCII Korean sets are similarly mapped and are:

S	1088	>	KO16MSWIN949	#	Korean KS single-byte PC-Data	ASCII
D	951	>	KO16MSWIN949	#	Korean KS double-byte PC-Data	ASCII
M	949	=	KO16MSWIN949	#	Korean KS multi-byte PC-Data	ASCII
MBC	949	=	1088 951	#	Korean KS multi-byte PC-Data	ASCII

Notice that the multi-byte set is a bidirectional map to KO16MSWIN949, while the single and double codepages are mapped one-way to KO16MSWIN949. Because only one bidirectional CCSID to Oracle Database Character Set entry for a given pair can exist, we directly map the multi-byte sets. And because the single-byte and double-byte CCSIDs are ostensibly subsets of KO16MSWIN949, we map them as one-way entries. Note that there is no "shift" attribute in any of these mappings. This is because ASCII-based sets do not use shift out/in encapsulation. Instead, ASCII-based sets use a different method (which does not use a shift out/in protocol) to identify double-byte characters.

The above entries supply the necessary codepage mappings for the gateway. To complete the example, we need to specify the correct character set in the HS_LANGUAGE and HS-NLS_NCHAR parameters in the gateway initialization file. The gateway initialization parameters would look as follows:

```
HS_LANGUAGE=AMERICAN_AMERICA.KO16MSWIN949
HS-NLS_NCHAR=KO16MSWIN949
```

Note that the specified character set must be ASCII-based.

This takes care of configuration of the gateway. The last step is to set up transforms between the EBCDIC codepages and the ASCII codepages in the DRDA Server database. Normally, the gateway would use a total of six transforms, one of each pair in both directions. You may save some table space by installing only the ASCII-to-EBCDIC transforms. The reasoning is that the DRDA Server needs to translate only the ASCII data that is sent by the gateway, but the DRDA Server does not need to send ASCII data. The gateway will receive the EBCDIC data and translate as needed. This one-sided data transfer methodology is called "receiver-makes-right", meaning that the receiver must translate whatever character set the sender uses. In our example, the DRDA Server is EBCDIC-based, so it will send all data in EBCDIC. The server, therefore, does not need to have an EBCDIC-to-ASCII transform because the server will never use the transform.

In our previous example, the DRDA Server database is assumed to be EBCDIC, which is likely to be true for a DB2/OS390 database. For a DB2/UDB database, however, this is not likely to be true. Because most DB2/UDB databases are running on ASCII-based computers, they will likely be created with ASCII-based codepages. In such cases, the gateway needs to have only one set of codepage map definitions, which are those for the ASCII set. Also, because both the DRDA Server and the gateway will be using the same codepages, no character set transforms need to be loaded into the DB2 database. This can help reduce the amount of cpu overhead that is associated with character translation.

One final note concerning codepage map entries: Be aware that some multi-byte codepages may be composed of single-byte CCSIDs that are already defined in the **codepage.map** file that is provided with the product. If you are adding a new set of entries to support a multi-byte set, then please comment out the provided entries so that your new entries will be used correctly.

Additional codepage mappings, which are not already provided, are possible. You may construct entries such as those in our examples, given knowledge of the IBM CCSIDs and the Oracle Database Character Sets. Because this can be complex and quite confusing (given the IBM documentation of codepage definitions and Oracle Database Character Set definitions), please thoroughly test your definitions for all desired character data values before putting them into production.

If you are uncertain, contact Oracle Support Services to request proper codepage mapping entries.

D.6 Message Availability

Whether a language message module is available depends on which modules are installed in the Oracle product set that is running on the server. If message modules for a particular language set are not installed, then specifying that language with a language parameter does not display messages in the requested language.

D.7 Example of NLS Configuration

Following is an example of all the settings that are needed to configure the gateway, DRDA Server, Oracle Database server, and client so that a language and character set are working compatibly across the system. In this example, the settings allow a customer in Germany to interact with the gateway in German:

Gateway *initsid.ora* file:

```
HS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P1
HS_NLS_DATE_FORMAT=YYYY-MM-DD
```

DRDA Server CCSID:

```
273 (D8EBCDIC273)
```

Oracle Database server and client setting for database:

```
SELECT USERENV('language') FROM DUAL;
USERENV('LANGUAGE')
-----
AMERICAN_AMERICA.WE8ISO8859P1
```

Oracle Database server and client environment variables:

```
NLS_LANG=GERMAN_GERMANY.WE8ISO8859P1
```

Configuration Worksheet

The table below is a worksheet that lists all of the parameter names and the reasons that you will need them for configuring the Gateway and the Communications Interfaces (SNA and TCP/IP). Use the worksheet to gather the specific information that you need before you begin the configuration process.

Table E-1 List of Parameters Needed to Configure the Gateway

Reason	Name of Parameter Needed	Your Specific Parameters Here
For: Gateway's Oracle Home	ORACLE_HOME	_____
For: Gateway's System ID	ORACLE_SID	_____
For: Primary Service Definition	SNA Network Name	_____
For: Primary Service Definition	Control Point Name	_____
For: Connection Properties Address	Remote Network Address	_____
For: Connection Properties Address	Remote SAP Address	_____
For: System Identification: Local Node Name and Remote Node Name	For Each: Network Name	_____
For: System Identification: Local Node Name and Remote Node Name	Control Point Name	_____
For: System Identification: Local Node Name and Remote Node Name	Local Node ID	_____
For: System Identification: Local Node Name and Remote Node Name	Remote Node ID	_____

Table E-1 (Cont.) List of Parameters Needed to Configure the Gateway

Reason	Name of Parameter Needed	Your Specific Parameters Here
For: Creating a Local LU Definition:	LU Alias	_____
For: Creating a Local LU Definition:	LU Name	_____
For: General APPC Mode Definition	Mode Name	_____
For: APPC Mode Limits	Parallel Session Limit	_____
For: APPC Mode Limits	Minimum Contention Winner Limit	_____
For: APPC Mode Limits	Partner Min Contention Winner Limit	_____
For: APPC Mode Limits	Automatic Activation Limit	_____
For: APPC Mode Characteristics	Pacing Send Count	_____
For: APPC Mode Characteristics	Pacing Receive Count	_____
For: APPC Mode Characteristics	Max Send RU Size	_____
For: Remote LU Definition, General Properties	Appropriate Connection name	_____
For: Remote LU Definition, General Properties	LU Alias	_____
For: Remote LU Definition, General Properties	Network Name	_____
For: Remote LU Definition, General Properties	Uninterpreted Network Name	_____
For: Remote LU Properties Options	Any Security Options needed	_____
For: Remote collection ID	DRDA_PACKAGE_COLLID	_____
For: Remote package name	DRDA_PACKAGE_NAME	_____
For: Creating CPI-C Symbolic Destination Names (Side Information Profiles), general information	Appropriate Name for each Side Information Profile	_____

Table E-1 (Cont.) List of Parameters Needed to Configure the Gateway

Reason	Name of Parameter Needed	Your Specific Parameters Here
For: Creating CPI-C Symbolic Destination Names (Side Information Profiles), general information	Appropriate Mode	_____
For: Partner Information in CPI-C Name Properties	TP Name	_____
For: Partner Information in CPI-C Name Properties	Partner LU Name Alias	_____
For: Configuring TCP/IP	Local Hostname, Domain Name	_____
For: Configuring TCP/IP	IP Address	_____
For: Configuring TCP/IP	Network Mask	_____
For: Configuring TCP/IP	Name Server IP Address	_____
For: Configuring TCP/IP	Destination Hostname or IP Address	_____
For: Configuring TCP/IP	Destination Service Port Number	_____
For: Recovery user ID	DRDA_RECOVERY_USERID	_____
For: Recovery Password	DRDA_RECOVERY_PASSWORD	_____
For: Remote Database Name	DRDA_REMOTE_DB_NAME	_____
For: Connection Parameter	DRDA_CONNECT_PARM	_____
For: Owner ID of DRDA package	DRDA_PACKAGE_OWNER	_____
For: DB Name used with Oracle Database server	HS_DB_NAME	_____
For: DB Domain used with Oracle Database server	HS_DB_DOMAIN	_____

Note: The user ID that is used to bind or rebind the DRDA package must have the appropriate privileges on the remote database as described in [Chapter 5, "Configuring the DRDA Server"](#). Your database administrator will need to provide these privileges.

Quick Reference to Oracle SQL Functions

The following is a list of the Oracle SQL functions in alphabetic order.

ABS
ACOS
ADD_MONTHS
ASIN
ASCII
ATAN
ATAN2
CEIL
CHAR_TO_ROWID
CHR
CONVERT
COS
COSH
DECODE
DUMP
EXP
FLOOR
GREATEST
HEXTORAW
INITCAP
INSTR
INSTRB
LAST_DAY
LEAST
LENGTH
LENGTHB
LN

LOG
LOWER
LPAD
LTRIM
MOD
MONTHS_BETWEEN
NEW_TIME
NEXT_DAY
NLS_INITCAP
NLS_LOWER
NLS_UPPER
NLSSORT
POWER
RAWTOHEX
REPLACE
ROUND
ROWIDTOCHAR
RPAD
RTRIM
SIGN
SIN
SINH
SOUNDEX
SQRT
STDDEV
SUBSTR
SUBSTRB
SYSDATE
TAN
TANH
TO_CHAR
TO_DATE
TO_LABEL
TO_MULTI_BYTE
TO_NUMBER
TO_SINGLE_BYTE
TRANSLATE

TRUNC
UID
UPPER
USER
USERENV
VARIANCE
VSIZE

Sample Applications

This appendix contains sample applications that can be used with the gateway:

- [DB2INS](#) on page G-2
- [ORAIND](#) on page G-4

G.1 DB2INS

DB2INS is a sample DB2 stored procedure that inserts a row into a DB2 table. This procedure uses the SIMPLE linkage convention.

```

/*****
/*
/* This DB2 stored procedure inserts values for the DNAME and LOC
/* columns of DB2 user table SCOTT.DEPT.
/*
/* The SCOTT.DEPT table is defined to DB2 as
/*     DEPTNO INTEGER, DNAME CHAR(14), LOC VARCHAR(13).
/*
/* This procedure receives 3 input parameters from the calling
/* program which contain the values to insert for DEPTNO, DNAME, and
/* LOC.
/*
/* The linkage convention used for this stored procedure is SIMPLE.
/*
/* The output parameter for this procedure contains the SQLCODE from
/* the INSERT operation.
/*
/* The entry in the DB2 catalog table SYSIBM.SYSPROCEDURES for this
/* stored procedure might look like this:
/*
/* INSERT INTO SYSIBM.SYSPROCEDURES
/* (PROCEDURE, AUTHID, LUNAME, LOADMOD, LINKAGE, COLLID, LANGUAGE,
/* ASUTIME, STAYRESIDENT, IBMREQD, RUNOPTS, PARMLIST)
/* VALUES
/* ('DB2INS', ' ', ' ', 'DB2INS', ' ', 'DB2DEV', 'C', '0', ' ',
/* 'N', ' ', 'A INT IN, B CHAR(14) IN, C VARCHAR(13) IN,
/* D INT OUT, E CHAR(10) OUT');
*****/
#pragma runopts(plist(os))
#include <stdlib.h>
#include <stdlib.h>
EXEC SQL INCLUDE SQLCA;
/*****
/* Declare C variables for SQL operations on the parameters. These
/* are local variables to the C program which you must copy to and
/* from the parameter list provided to the stored procedure.
*****/
EXEC SQL BEGIN DECLARE SECTION;
long dno;          /* input parm - DEPTNO */
char dname[15];    /* input parm - DNAME */
char locale[14];   /* input parm - LOC */
EXEC SQL END DECLARE SECTION;
main(argc,argv)
int argc;
char *argv[];
{
/*****
/* Copy the input parameters into the area reserved in the local
/* program for SQL processing.
*****/
dno = *(int *) argv[1];
strcpy(dname, argv[2]);
strcpy(locale, argv[3]);
/*****
/* Issue SQL INSERT to insert a row into SCOTT.DEPT
*****/

```

```
EXEC SQL INSERT INTO SCOTT.DEPT VALUES(:dno, :dname, :locale);
/*****
/* Copy SQLCODE to the output parameter list.          */
*****/
*(int *) argv[4] = SQLCODE;
}
```

G.2 ORAIND

ORAIND is a sample host program that calls a DB2 stored procedure (DB2INS) to insert a row into a DB2 table.

```

/*****
/* This sample ProC program calls DB2 stored procedure DB2INS to
/* insert values into the DB2 user table SCOTT.DEPT. This calling
/* program uses embedded PL/SQL to call the stored procedure.
*****/
#include <stdio.h>EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR      username[20];
    VARCHAR      password[20];
    int          dept_no;
    char         dept_name[14];
    VARCHAR      location[13];
    int          code;
    char         buf[11];
    int          x;
EXEC SQL END DECLARE SECTION;
EXEC SQL INCLUDE SQLCA;
main()
{
/*****
/* Setup Oracle userid and password
*****/
    strcpy(username.arr, "SCOTT");          /* copy the username */
    username.len = strlen(username.arr);
    strcpy(password.arr, "TIGER");          /* copy the password */
    password.len = strlen(password.arr);
    EXEC SQL WHENEVER SQLERROR GOTO sqlerror;
/*****
/* Logon to Oracle
*****/
    EXEC SQL CONNECT :username IDENTIFIED BY :password;
    printf("\nConnected to ORACLE as user: %s\n", username.arr);
    /* Delete any existing rows from DB2 table */
    EXEC SQL DELETE FROM SCOTT.DEPT@GTWLINK;
    EXEC SQL COMMIT;
/*----- begin pl/sql block -----*/
/*****
/* Insert 1 row into DB2 table SCOTT.DEPT by invoking DB2 stored
/* procedure DB2INS. The DB2 stored procedure will perform the
/* INSERT.
/*
/* SCOTT.DEPT table is defined on DB2 as:
/*
/*   DEPTNO    INTEGER;
/*   DNAME     CHAR(14);
/*   LOC       VARCHAR(13);
/*
*****/
    EXEC SQL EXECUTE BEGIN      :dept_no := 10;
        :dept_name := 'GATEWAY';
        :location := 'ORACLE';
        DB2INS@GTWLINK(:dept_no, :dept_name, :location, :code);
    END;
    END-EXEC;
/*----- end pl/sql block -----*/

```

```

/*****
/* Check the SQLCODE returned from the stored procedures INSERT.          */
*****/
if (code == 0)
    printf("DB2INS reports successful INSERT\n");
else
{
    printf("DB2INS reports error on INSERT.\nSQLCODE=%d\n",code);
    goto sqlerror
}
/*****
/* Verify row insertion.  Query the data just inserted.                  */
*****/
EXEC SQL SELECT deptno, dname, loc INTO
    :dept_no, :dept_name, :location
    FROM SCOTT.DEPT@GTWLINK WHERE deptno = 10;
printf("\nData INSERTed was:\n");
printf("\ndeptno = %d, dname = %s, loc = %s\n",
    dept_no, dept_name, location.arr)
/*****
/* Logoff from Oracle                                                    */
*****/
EXEC SQL COMMIT RELEASE;
printf("\n\nHave a good day\n\n");
exit(0);
sqlerror:
    printf("\n% .70s \n", sqlca.sqlerrm.sqlerrmc);
    EXEC SQL WHENEVER SQLERROR CONTINUE;
    EXEC SQL ROLLBACK RELEASE;
    exit(1);
}

```


Symbols

\$DISPLAY

environmental variables, 7-3, 9-4

\$ORACLE_HOME, 7-3, 9-3

A

accessing

DRDA servers, 12-9

gateway, 12-9

main topic, 13-3

Advanced Program to Program Communication
(APPC/LU6.2) protocol, 6-1, 7-1, 8-1, 9-1

Advanced Security

CHECKSUM command, 11-5

encryption, 11-5

international version types supported, 11-6

resetting configuration parameters on

gateway, 11-7

setting test parameters for gateway, 11-6

setting test parameters for Oracle integrating
server, 11-6

setting up for test, 11-6

testing gateway and Oracle integrating
server, 11-6

export encryption algorithms, 11-5

function of the gateway, 1-5

purpose, 1-5

test error, 11-6

12660, 11-6

AGW ADD USERID command, 15-4

AGW DELETE USERID command, 15-4

AIX

AIX_RS6K, default value for

DRDA_LOCAL_NODE_NAME, C-7

System Management Interface Tool, 8-2

user ID for DB2/UDB, 5-6

alias

CPI-C destination, 7-12, 9-12

DB2, C-4

Ethernet parameters, 9-8

local LU alias, 7-9

partner LU, 8-3

alias objects, DB2

known restrictions, 2-4

ALL_CATALOG view, A-3

ALL_COL_COMMENTS view, A-3

ALL_CON_COLUMNS view, A-3

ALL_CONSTRAINTS view, A-4

ALL_DB_LINKS data dictionary view, 13-3

ALL_INDEXES view, A-4

ALL_OBJECTS view, A-6

ALL_SYNONYMS view, A-7

ALL_TAB_COMMENTS view, A-10

ALL_TABLES view, A-7

ALL_USERS view, A-10

ALL_VIEWS view, A-10

allocation mode, SNA session,

DRDA_CMSRC_CM_IMMEDIATE, C-4

ALTER session statement, 13-2

ANSI-standard SQL, 1-5, 1-12

API (application program interface), 11-3

APPC

concurrent connections, 3-2

configuring IBM Communication Server, 8-1

configuring SNAP-IX interfaces, 7-1

configuring SUNLINK, 6-1

conversations, 8-2

creating SNAP-IX profiles, 7-2

creating SNAPplus2 profiles, 9-2

DB2/VM, 15-4

number of database link connections, 14-6

password length, 17-5

user ID length, 17-5

APPC VTAM Support (AVS)

also see AVS, 15-4

known restrictions, 2-4

APPC/LU6.2

parallel session limits, 8-3

protocol, 7-1, 9-1

side information profile, 8-4

APPEND command

supported by COPY, 13-7

application

authenticating logons, 15-2

portability, 1-12

server support, 1-4

application development on the gateway, 1-14

application program interface (API), 11-3

APPN

node, advanced peer-to-peer networking

- node, 8-4
- support requires VTAM Version 4, 8-4
- architecture of the gateway, 1-8
- array size
 - fetch reblocking, 1-11
 - how determined, 14-2
- AS/400
 - command, DSPRDBDIRE, 5-5
 - communications, configuring, 5-4
 - files and file members, accessing, 13-3
 - library name, DRDA_PACKAGE_COLLID, C-7
- ASCII
 - code page, D-4
 - sort order, 14-21
 - tables, known restrictions, 1-14
 - translated from EBCDIC, 14-23
 - US7ASCII, D-6
 - US7ASCII as default, D-4
- authority
 - CONNECT, 15-5
 - execute, 15-5
- autonomy, site, 1-7
- available products menu, 4-6
- AVS
 - also see APPC VTAM Support, 2-4
 - configuring, 5-7
 - DB2/VM, 15-4
 - mapping user IDs, 2-4, 15-4
 - mapping user IDs (DB2/VM)
 - known restrictions, 2-4
 - mapping user IDs with DB2/VMT, 15-4

B

- binary data, non-character, 14-23
- Bind Package Stored Procedure
 - DB2/400, 5-4
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
- BIND privilege
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
- bind variables
 - known restrictions, SQL limitations, 2-6
 - SQL passthrough, 14-28
- BINDADD authority, 12-6
- BINDADD privilege
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
- BINDAGENT privilege
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
- binding the DRDA package
 - backward compatibility, 16-4
 - DB2/400, 5-4
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6

- DB2/VM, 5-7
 - on DB2/UDB, 12-6
 - on DRDA server, 12-5
- pre-binding checklist, 12-7
- Bourne shell
 - setting ORACLE_HOME variable, 4-4
- bug
 - debugging, 13-7, 17-3, 17-6, 17-7
 - number 205538, known restrictions, SQL
 - limitations, 2-6
 - number 3275652, 2-3

C

- C shell
 - setting ORACLE_HOME variable, 4-4
- call
 - a CICS or IMS transaction, 14-6
 - DB2 stored procedure, 14-4, 14-5
 - DRDA_DISABLE_CALL, 12-7, B-2
 - empproc stored procedure, 14-4
 - Oracle Call Interfaces, 17-2
 - PL/SQL, 14-5
 - stored procedure, 14-3
 - to stored procedure
 - known restrictions, 2-4
- capabilities of DRDA server, native semantics, 14-19
- CCSID
 - 65535 as the default for all tables created, C-4
 - codepage mapping facility, D-7
 - description, D-3
 - external mapping to Oracle character sets
 - supported, 1-14
- CCSID (coded character set identifiers),
 - defined, D-10
- CD-ROM drive requirements, 3-2
- changed parameters, 16-3
- changes in this release
 - IBM DB2 Version 5.1 EBCDIC and ASCII
 - Tables, 1-14
 - IBM DB2/UDB supported, 1-14
 - read-only support, 1-14
- CHARACTER SET clause, client/server
 - configuration, D-4
- CHARACTER SET parameter
 - description, D-2
- character sets
 - and codepage map facility, D-7
 - ASCII, 1-14
 - CCSID, D-3
 - codepage, C-4
 - EBCDIC, 1-14
 - supported, 1-14
- character string
 - converting datatypes, 14-23
 - performing operations, 14-23
- checklist
 - configuring the communications interfaces, 6-2, 8-2
 - configuring the DRDA server, 5-2

- gateway configuration, 12-2
- gateway installation, 4-3
- installation overview, 4-3
- Oracle Net, 11-2
- CHECKSUM command, 1-5, 11-5
- CICS transaction, 14-6
- clause
 - CHARACTER SET, client/server configuration, D-4
- clauses
 - CONNECT TO, 13-2
 - GROUP BY, SQL Set Clauses, 14-21
 - HAVING, SQL Set Clauses, 14-21
 - ORDER BY, SQL Set Clauses, 14-21
 - SQL
 - DELETE, 14-26
 - INSERT, 14-26
 - SELECT WHERE, 14-26
 - UPDATE, 14-26
 - USING, 13-2
 - VALUES
 - functions not allowed by DB2, 14-26
 - WHERE
 - known restrictions, SQL limitations, 2-6
 - WHERE CURRENT OF CURSOR
 - known restrictions, SQL limitations, 2-6
 - WHERE, SQL Set Clauses, 14-21
- code tracing, C-13
- codepage map facility
 - configuring support for character sets, known restrictions, 2-4
 - for data translation, D-7
 - supported by gateway, 1-14
- coercion
 - of data, 14-19
- collection privilege - CREATE IN, 5-3, 5-7
- collection privilege - CREATETAB, 5-3
- column
 - date columns, TO_DATE function, 14-26
 - Oracle ROWID
 - known restrictions, 2-6
 - supported in a result set, 1-12
- commands
 - AGW ADD USERID, 15-4
 - AGW DELETE USERID, 15-4
 - CHECKSUM, 1-5, 11-5
 - COPY
 - known restrictions, 2-4
 - Oracle Database server to DRDA server, 13-6
 - COPY (SQL*Plus command), 13-7
 - CREATE DATABASE LINK
 - DB2/VM, 5-7
 - DESCRIBE
 - known restrictions, 2-3
 - EXECUTE, 1-6
 - EXPLAIN PLAN, 17-6
 - INSERT
 - known restrictions, 2-4
 - INSERT, not supported, 13-6
 - lanscan, 7-6, 9-6
 - commit confirm protocol, 1-6
 - Communication Database (CDB) tables, DDF, 5-4
 - communications requirements, 3-3
 - Communications Server
 - configuring server profiles, 8-2
 - profiles
 - creating, 8-2
 - compatible SQL set operators and clauses, 14-21
 - concatenation restrictions, 2-5
 - concurrent connections
 - APPC, 3-2
 - TCP/IP, 3-2
 - configuration
 - SNA sample file, 6-3
 - configuring
 - additional DRDA servers, 12-9
 - AS/400 communications, 5-4
 - AVS, 5-7
 - binding DRDA package, 12-7
 - checklist for gateway, 12-2
 - checklist for installation overview, 4-3
 - checklists, 4-3, 5-2, 6-2, 8-2, 11-2, 12-2
 - communications interfaces, steps for, 7-2, 9-2
 - DB2/400, 5-4, 5-5
 - DB2/OS390, 5-3
 - DB2/UDB, 5-5
 - host workstation for gateway, 12-4
 - MVS VTAM, 5-3
 - Oracle integrating server, 12-8
 - Oracle Net, 11-4
 - other Oracle Database servers, 12-9
 - SNA server
 - creating profiles, 8-2
 - testing connection, 8-5
 - SNAPLUS2 on HP-UX, 9-1
 - SunLink SNA Peer-to Peer product, 6-3
 - TCP/IP
 - DB2/400, 5-4
 - DB2/OS390, 5-3
 - DB2/UDB, 5-5
 - DB2/VM, 5-7
 - under Solaris, 10-2
 - VM VTAM, 5-7
 - CONNECT authority, 12-6, 15-5
 - CONNECT BY not supported
 - known restrictions, SQL limitations, 2-6
 - CONNECT privilege
 - DB2/UDB, 5-6
 - CONNECT TO clause, 13-2
 - contact information
 - in configuring SNAPLUS2, 7-7, 9-7
 - conversion
 - errors, C-4
 - convert
 - character string, 14-23
 - datatypes
 - DRDA to Oracle datatypes, 14-22
 - DATE, 14-24
 - floating point to integer, 14-27
 - inbound user ID, 15-4

- into most suitable datatype, 14-27
- SQL, 1-10
- to the numeric datatype, 14-27
- converter, protocol, 1-4
- COPY command
 - Oracle Database server to DRDA server, 13-6
- COPY privilege
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
- COPY SQL*Plus command, 13-7
 - substituted for INSERT, known restrictions, 2-4
- copying data
 - from the DRDA server, 13-7
 - from the Oracle server to DRDA server, 13-6
- COS SQL function, 14-7
- COUNT function, 14-27
- CP (Control Point for the Node), 7-2, 9-2
- CPI-C
 - communication error messages, 17-3, 17-6
 - create Side Information Profile, 7-11, 9-11
 - profiles, creating, 7-10, 9-10
 - requires Side Information File, 6-4
 - routine, 17-3
 - side information profile, 7-11, 9-11
- CREATE command
 - supported by COPY, 13-7
- CREATE DATABASE LINK command, 13-2
 - DB2/VM, 5-7
- CREATE DATABASE statement, client/server
 - configuration, D-4
- CREATE IN privilege
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
- CREATE PUBLIC DATABASE LINK privilege, 12-5, 12-8
- CREATE TABLE statement, 1-5
- CREATEIN privilege
 - DB2/UDB, 5-6
- CREATETAB authority, 12-6
- CREATETAB privilege
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
- creating
 - a new directory, 4-4
 - CPI-C and mode definitions, 7-10, 9-10
 - database link, 13-2
 - link station, 7-6, 9-6
 - local LU names, 7-9, 9-8
 - partner LUs, 7-9, 9-9
 - remote node definition, 7-6, 9-6
 - SNA definitions for the gateway, 7-3, 9-3
 - SNAPLUS2 profiles, 7-2, 9-2
- cursor
 - defining the number of, 14-30
 - number of cursors,
 - DRDA_PACKAGE_SECTIONS, C-9
 - stability, DRDA_ISOLATION_LEVEL, C-6

D

- data coercion, 14-19
- data control language (DCL), 1-5
- DATA datatype, 14-24
- data definition language (DDL), 1-5
- data dictionary
 - support, 12-7
 - using, 14-30
 - views
 - ALL_DB_LINKS, 13-3
 - considerations for migration from previous releases, 16-4
 - emulation on DRDA server, 14-30
 - for DB2/VM and DB2/UDB not supported, A-2
 - list and descriptions, A-3
 - supported for DB2/OS390 and DB2/400 servers, A-2
 - USER_DB_LINKS, 13-3
- database
 - catalogs, 14-30
- database authorities - CONNECT, BINDADD, and CREATETAB, 5-6
- database link
 - behavior, 14-6
 - binding the gateway package, 12-5
 - creating, 13-2
 - defining and controlling, 15-3
 - dropping links, 13-2
 - examining, 13-3
 - guidelines, 13-2
 - limits, 13-3
 - processing, 13-2
 - public, 12-8
 - suffix, 14-2
 - to identify the gateway, 1-10
- database triggers, 1-4
- datatype
 - character string, 14-23
 - column (ALL_TAB_COLUMNS), A-9
 - column (USER_TAB_COLUMNS), A-18
 - conversion
 - DRDA to Oracle datatypes, 14-22
 - no control over, 14-27
 - converting character string, 14-23
 - data and time, 14-24
 - differences between Oracle server and DRDA databases, 14-1
 - DRDA server datatypes list, 14-22
 - mapping, 14-22
 - numeric, 14-27
 - operations, numeric, 14-27
 - Oracle datatypes RAW and LONG RAW, 14-23
 - restrictions, 14-22
 - size and value limitations, 14-22
- datatypes
 - DATE, 14-24
 - GRAPHIC, 14-23
 - LONG, 14-23
 - LONG RAW, 14-23

- Oracle and IBM DATE, 14-24
- Oracle DATE, 14-24
- RAW, 14-23, C-4
- TIME, 14-24
- TIMESTAMP, 14-24
- VARCHAR, 14-23
- date
 - date columns, TO_DATE function, 14-26
 - DELETE
 - statement, 14-25
 - HS_NLS_DATE_FORMAT parameter, 14-25, 14-26
 - INSERT
 - statement, 14-25
 - operations, 14-24
 - SELECT statement, 14-25
 - TO_DATE function, 14-25
 - UPDATE
 - statement, 14-25
- date arithmetic
 - known restrictions, 2-5
- DATE datatype, 14-24
- DB_DOMAIN parameter
 - known restrictions, 2-5
- DB2
 - 02pcg.sql granting authority, 12-8
 - alias objects
 - known restrictions, 2-4
 - aliases, C-4
 - CICS, 14-6
 - configuring DB2/OS390, 5-3
 - data access, 1-6
 - Distributed Data Facility (DDF), 5-4
 - DRDA_DESCRIBE_TABLE compatibility, C-4
 - IBM DB2 Version 5.1 ASCII Tables, 1-14
 - IMS, 14-6
 - native SQL, 1-5
 - native stored procedures, 1-6
 - procedural feature considerations, 14-6
 - SPUFI utility, 5-4
 - SQL statements, 14-28
 - statements
 - CREATE TABLE, 1-5
 - stored procedures, 14-6
- DB2/400
 - catalog view, 14-30
 - configuring, 5-4, 5-5
 - data dictionary views supported by gateway, A-2
 - DRDA_DEFAULT_CCSD, C-4
 - DRDA_DISABLE_CALL, C-5
 - DRDA_ISOLATION_LEVEL, C-6
 - DRDA_OPTIMIZE_QUERY, C-7
 - DRDA_PACKAGE_COLLID, C-7
 - user ID mapping, 15-5
- DB2/OS390
 - catalog view, 14-30
 - configuring, 5-3
 - data dictionary views supported by gateway, A-2
 - DRDA_DISABLE_CALL, C-5
 - DRDA_ISOLATION_LEVEL, C-6
 - DRDA_OPTIMIZE_QUERY, C-7
 - user ID mapping, 15-4
- DB2/OS390 V6 and V7
 - stored procedures supported, 1-14
- DB2/UDB
 - binding packages, 12-6
 - binding the gateway package, 12-7
 - catalog view, 14-30
 - configuring, 5-5
 - data dictionary views not supported, A-2
 - DRDA_ISOLATION_LEVEL, C-6
 - DRDA_OPTIMIZE_QUERY, C-7
 - incompatibilities with gateway, 2-3
 - known restrictions, 2-5
 - supported, 1-14
 - user ID mapping, 15-5
- DB2/VM
 - catalog view, 14-30
 - configuring, 5-7
 - data dictionary views not supported, A-2
 - database and SQL functions, 14-16
 - DRDA_ISOLATION_LEVEL, C-6
 - DRDA_OPTIMIZE_QUERY, C-7
 - DRDA_PACKAGE_OWNER, C-8
 - instance, DRDA location name, 5-7
 - server machine, 15-5
 - user ID mapping, 15-4
- DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function, 14-28
- DD basic tables, known restrictions, 2-4
- DDF
 - DB2 (Distributed Data Facility), 5-4
 - subsystem, 5-3
- DDL
 - statement, 14-28
- debug
 - gateway, 17-6, 17-7
 - library, 17-7
- debugging
 - error codes, 17-3
 - SQL tracing, 17-6, 17-7
 - your application, 13-7
- de-installing the gateway, 4-7
- DELETE
 - known restrictions, SQL limitations, 2-6
 - operation, 14-2
 - read-only gateway, 13-6
 - SQL clause, 14-26
 - statement, 14-28
- DESCRIBE
 - character string operations, 14-23
 - command, known restrictions, 2-3
- diagnostic parameter, C-12
- dictionary
 - mapping, 1-5
 - tables, 14-30
- DICTIONARY view, A-11
- disk space requirements, 3-3
- distributed
 - operations, DB2, 5-4

- queries
 - example of, 13-4
 - two-phase commit, 13-5
- transaction, DRDA_RECOVERY_USERID, C-9
- distributed applications
 - support for, 1-13
- distributed data facility (DDF), 5-4
- distributed database, 11-3
- distributed DRDA transactions, 13-5
- distributed processing, 11-3
- distributed query optimizer (DQO), 13-4
 - DRDA-specific parameters, C-7
- dmesg, 7-6
- double-byte support, D-10
- DQO
 - also see distributed query optimizer, 13-4
 - DRDA-specific parameters, C-7
- drc error code, 17-3
- DRDA
 - catalog, 14-30
 - database requirements, 3-3
 - defining number of cursors, 14-30
 - gateway package considerations, 12-7
 - location name
 - for DB2/UDB instance, 5-6
 - for DB2/VM instance, 5-7
 - mode
 - IBMRDB, 7-11, 9-10
 - session security options, 15-4
- DRDA Application Server Function, 1-12
- DRDA server
 - accessing, 12-9
 - and dependent LUs, 7-2, 9-2
 - and independent LUs, 7-2, 9-2
 - and Side Information Profiles, 7-2, 9-2
 - architecture, 1-8
 - capabilities, native semantics, 14-19
 - character sets
 - known restrictions, 2-4
 - client applications initiate conversations using
 - dependent LUs, 7-2, 9-3
 - configuring
 - DB2/400, 5-4
 - DB2/OS390, 5-3
 - DB2/UDB, 5-5
 - DB2/VM, 5-7
 - considerations for binding packages, 16-4
 - copying data
 - from Oracle server, 13-6
 - to Oracle server, 13-7
 - database link behavior, 14-6
 - default TP name, 7-12, 9-12
 - functions, 14-19
 - languages and character sets in
 - configuration, D-7
 - Logical Unit (LU), 6-5, 7-13, 8-5, 9-13
 - partner LU, 7-9, 9-9
 - stored procedures, 14-4
 - VTAM definitions and local LU names, 7-9, 9-8
- DRDA_CAPABILITY parameter, 14-19

- DRDA_CMSRC_CM_IMMEDIATE parameter, C-4
- DRDA_CODEPAGE_MAP parameter, D-7
- DRDA_COMM_BUFLLEN parameter, C-3
- DRDA_CONNECT_PARM (SNA format)
 - parameter, C-3
- DRDA_CONNECT_PARM (TCP/IP format)
 - parameter, C-3
- DRDA_CONNECT_PARM parameter, 17-3
 - determine SNA V9 value for worksheet, 6-5
- DRDA_DEFAULT_CCSID parameter, C-4
- DRDA_DESCRIBE_TABLE parameter, C-4
 - known restrictions, 2-4
- DRDA_DESCRIBE_TABLE=FALSE initialization
 - parameter
 - known restrictions, 2-4
- DRDA_DISABLE_CALL parameter, 12-7, C-5
- DRDA_FLUSH_CACHE parameter, C-5
- DRDA_GRAPHIC_LIT_CHECK parameter, C-6
- DRDA_GRAPHIC_PAD_SIZE parameter, C-5
- DRDA_GRAPHIC_TO_MBCS parameter, C-6
- DRDA_ISOLATION_LEVEL parameter, C-6
- DRDA_LOCAL_NODE_NAME parameter, C-7
- DRDA_MBCS_TO_GRAPHIC parameter, C-7
- DRDA_OPTIMIZE_QUERY parameter, 13-4, C-7
- DRDA_PACKAGE_COLLID parameter, 17-4, C-7
- DRDA_PACKAGE_CONSTOKEN parameter, C-8
- DRDA_PACKAGE_NAME parameter, 12-7, 17-4, C-8
- DRDA_PACKAGE_OWNER parameter, C-8
- DRDA_PACKAGE_SECTIONS parameter, 14-30, C-9
- DRDA_READ_ONLY parameter, 13-5, C-9
- DRDA_RECOVER_USERID, 5-6
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
- DRDA_RECOVERY_PASSWORD parameter, 5-6, C-9
 - DB2/400, 5-5
 - DB2/OS390, 5-3
- DRDA_RECOVERY_USERID parameter, 5-6, C-9
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
- DRDA_REMOTE_DB_NAME parameter, C-10
- DRDA_SECURITY_TYPE parameter, C-10
- DROP DATABASE LINK statement, 13-2
- dropold.sql script, 12-8
- DSPRDBDIRE command, 5-5
- dynamic dictionary mapping, 1-5

E

- EBCDIC
 - character set support, D-6
 - code page, D-4
 - DRDA server CCSID, D-12
 - sort order, 14-21
 - tables, known restrictions, 1-14
 - translated to ASCII, 14-23

- EMP
 - system-wide synonym, 13-4
 - table, 13-6
- empproc
 - stored procedure, 14-4
- encryption types for Advanced Security, 11-6
- environment
 - heterogeneous, 13-6
- environment variable
 - NLS_LANG, D-4
- environmental variable
 - NLS_LANG, D-3
 - ORA_NLS33, D-2
 - ORACLE_HOME, 4-4
- environmental variables
 - \$DISPLAY, 7-3, 9-4
- errd
 - mapped error example, 17-4
- errmc
 - CPI-C routine, 17-3
 - errmc field lists any error tokens, 17-3
 - errmc=1245, error at gateway bind, Bug 3275652, 2-3
 - ERRMC=124C, Bug 3121041 (fixed), 2-2
 - error tokens, 17-3
 - mapped error example, 17-4
- errp
 - errp field indicates program that detected error, 17-3
- errnc
 - ERRNC 124C, Bug 2787316 (fixed), 2-2
- error
 - 12660 (test error for Advanced Security), 11-6
 - basic description, 17-2
 - change, ORA-09100 to ORA-28500, 17-3
 - change, ORA-09101 to ORA-28501, 17-3
 - codes
 - drc, 17-3
 - grc, 17-3
 - communication, 17-3
 - condition, 17-2
 - conversion, C-4
 - date, D-6
 - detected
 - by Oracle integrating server, 17-2
 - by server database, 17-3
 - by the gateway, 17-2
 - in DRDA software, 17-3
 - in the DRDA software, 17-3
 - drc= field
 - 300xx, 17-5
 - 7xx, 17-5
 - HGO-00706, 17-2
 - host database, 17-5
 - interpreting error messages, 17-2
 - mapped sqlstate, 17-4
 - messages, 6-6, 7-14, 8-6, 9-14, 14-23
 - messages & codes, 17-1
 - number, 17-2
 - obsolete parameters, 16-4

- ORA-00001, index constraint violated, 17-4
- ORA-00942
 - mapped error example, 17-4
 - object name too long, 17-4
- ORA-01017, logon denied, 17-4
- ORA-01031, insufficient privileges, 17-4
- ORA-01460, invalid CCSID, 17-4
- ORA-01476, divide by zero, 17-4
- ORA-02019, 17-2
- ORA-2025, when using INSERT command, 13-6
- ORA-28500 (was ORA-09100), 17-3
- ORA-28501
 - communication error, 17-3
 - was ORA-09101, 17-3
- ORA-9100 to ORA-9199, 17-2
- Oracle mapped error codes, 17-4
- specific gateway error codes, 17-5
- tokens, 17-3
- translation, 14-23
- while binding the gateway package, 12-6
- with Native Semantics, 14-19
- error array (errd), See errd
- errp
 - errp=GJDMRC, error at gateway bind, Bug 3275652, 2-3
 - ERRP=GJDMRCCH, Bug 3121041 (fixed), 2-2
 - mapped error example, 17-4
- EXCEPT set operator, SQL Set Clauses, 14-21
- execute authority, 15-5
- EXECUTE command, 1-6
- EXECUTE privilege
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
- exits
 - gateway local date, 14-26
- EXPLAIN PLAN command, 17-6
- EXPLAIN_PLAN table, 1-12
- export encryption algorithms, 11-5

F

- FDS_CLASS parameter, C-10
- FDS_CLASS_VERSION parameter, C-10
- FDS_INSTANCE parameter, C-11
- features
 - Oracle Snapshots, 13-6
- features of the gateway
 - application development and end-user tools, 1-14
 - application portability, 1-12
 - columns supported in a result set, 1-12
 - distributed applications supported, 1-13
 - EXPLAIN_PLAN improvement, 1-12
 - fetch reblocking, 1-11
 - heterogeneous database integration, 1-12
 - heterogeneous services architecture, 1-11
 - large base of data access, 1-12
 - main topic, 1-11
 - minimum impact on existing systems, 1-12
 - Native Semantics, 1-12

- Oracle Database passthrough supported, 1-11
- performance enhancements, 1-11
- remote data access, 1-13
- retrieving result sets through passthrough, 1-11
- support for TCP/IP, 1-12
- fetch array size, with HS_FDS_FETCH_ROWS, C-11
- fetch reblocking, 1-11, 14-2
- fetch date, C-12
- fields
 - errmc, lists any error tokens, 17-3
 - errp, indicates program that detected error, 17-3
- file member name, 13-3
- file members
 - accessing AS/400 files, 13-3
- files
 - initsid.ora
 - communication errors, 17-4
 - gateway error -700, 17-5
 - listener.ora, 11-5
 - sample, B-1
 - listener.ora, B-4
 - tnsnames.ora, B-3
 - sna_domn.cfg, 7-3, 9-3
 - sna_node.cfg, 7-3, 9-3
 - tnsnames.ora, 12-8, 12-9
 - modifying, 11-5
 - VSAM, 14-6
- FOR BIT DATA
 - DRDA_DEFAULT_CCSD, C-4
 - option, 14-23
- free session,
 - DRDA_CMSRC_CM_IMMEDIATE, C-4
- functions
 - COS, 14-7
 - COUNT, 14-27
 - DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE, 14-28
 - DRDA server, 14-19
 - SQL
 - SUBSTR, 14-19
 - SUBSTR
 - known restrictions, 2-4
 - TO_DATE, 14-24, 14-25, 14-26
 - TO_DATE, main topic, 14-26

G

- g4ddtab.sql script, 12-8
- g4ddview.sql script, 12-8
- gateway

- accessing
 - from other Oracle servers, 12-9
 - main topic, 13-3
- advantages
 - main topic, 1-3
 - migration and coexistence, 1-7
 - multi-site transactions, 1-6
 - security, 1-7
 - server technology and tools, 1-6
 - site autonomy, 1-7

- two-phase commit, 1-6
- and Oracle tools, 1-10
- and stored procedures (Oracle and non-Oracle), 1-5
- application tools, 1-14
- architecture, 1-8
- authenticating logons, 15-2
- benefits of integration with Oracle Database server, 1-4
- binding DRDA packages, 12-5
- components, 1-9
- configuration, 12-4
- definition of terms, 1-7
- de-installing, 4-7
- DRDA package considerations, 12-7
- error codes, 17-5
- errors detected, 17-2
- features, 1-11
 - main topic, 1-11
- how to access, 13-3
- installation
 - log file, 4-6
 - steps, 4-4
 - steps via Oracle Universal Installer, 4-6
- interface, 1-10
- local date exit, 14-26
- logging, LOG_DESTINATION, C-12
- migration problems, 16-2
- name, setting up, with Sunlink, 6-3
- parameter, 5-7
- performance, 14-19
- performance enhancements, 1-11
- performing distributed queries, 13-4
- read-only option, 13-5
- service name entries in the tnsnames.ora, 16-4
- settings for security options, 6-5, 7-13, 8-5, 9-13
- SQL differences, 1-10
- supported languages, D-7
- tracing SQL statements, 13-7
- tracing, LOG_DESTINATION, C-12
- using, 13-1
- with other Oracle products
 - SQL*Plus, 1-6
- Gateway Initialization File
 - backup and recovery of gateway
 - configuration, 12-8
 - configuring for binding the DRDA gateway package, 12-5
 - defining userids and passwords, 15-5
 - encrypting passwords, 15-6
 - if errors are reported, 12-6
 - modifying, C-2
 - parameters
 - DRDA_READ_ONLY, 13-5
 - LOG_DESTINATION, 17-7
 - LOG_DESTINATION parameter, tracing, 17-8
 - new since V4 gateway, 16-3
 - ORACLE_DRDA_TCTL, 17-7
 - stored in initsid.ora, C-2
 - TRACE_LEVEL, 17-7

- parameters, list, C-2
- sample, B-2
- Gateway System Identifier (SID), 12-4
- GCS virtual machine, 15-4
- GLOBAL_NAMES
 - known restrictions, 2-5
- GRANT, 15-5
- GRANT statement, 12-9
- granting authority to a package for DB2, 12-8
- GRAPHIC datatype, 14-23
- graphic string operations
 - unsupported, 14-23
- grc error code, 17-3
- GROUP BY clause
 - SQL Set Clauses, 14-21
- GTW\$_BIND_PKG
 - internal stored procedure, 12-7
 - stored procedure, 12-5

H

- hardware requirements, 3-2
- HAVING clause
 - SQL Set Clauses, 14-21
- heterogeneous database integration, 1-12
- Heterogeneous Services (HS), see HS, 1-2
- HGO-00706 error, 17-2
- host
 - DRDA server, SNA and TCP/IP protocols, 1-8
 - networking needs, 3-4
 - performing character string operations on, 14-23
 - relationship to gateway and Oracle Server, 1-8
 - security validation required to begin database
 - link, 6-5, 7-13, 8-5, 9-13
 - SNAP-IX definitions, 7-3
- host database error, 17-5
- host Logical Unit (LU), 6-5, 7-13, 8-5, 9-13
- host variable, 14-22, 14-23
- HP9000 host, 9-3, 9-8
 - invoking xsnapadmin, 9-4
- HS (Heterogeneous Services)
 - architecture features, 1-11
 - Oracle Net considerations, 16-4
- HS= (TNSNAMES parameter for Oracle Net), 11-5
 - gateway migration problems, 16-2
- HS_DB_DOMAIN parameter
 - known restrictions, 2-5
- HS_FDS_FETCH_ROWS parameter, C-11
- HS_LANGUAGE parameter, C-11
- HS_NLS_DATE_FORMAT, D-6
 - parameter, 14-26
- HS_NLS_DATE_LANGUAGE, D-6
- HS_NLS_NCHAR, D-6
- HS_NLS_NCHAR parameter, C-11
- HS_RPC_FETCH_REBLOCKING parameter, 1-11, 14-2
- HS_RPC_FETCH_SIZE parameter, 1-11, 14-2

I

- IBM Communications Server for AIX
 - and SNA security option, 6-6, 7-14, 8-6, 9-14
 - Version 5 required patches, 3-4
- IBMRDB, 7-12, 9-12
 - DRDA mode, 7-11, 9-10
- implementation, 1-9
- implicit data conversion, 14-19
- implicit protocol conversion, 1-4
- IMS transaction, 14-6
- IN and OUT columns, multi-byte support, D-10
- inbound connections
 - processing, 15-3
- initdrdahoal.ora
 - sample, B-2
 - see also Gateway Initialization File parameters
- initialization parameters
 - new since V4 gateway, 16-3
 - see also Gateway Initialization Files, parameters
- initsid.gtwboot file
 - migrating, 16-2
- initsid.ora file
 - communication errors, 17-4
 - gateway error -700, 17-5
 - migrating, 16-2
 - NLS parameters, D-5
- input bind variables, 14-25
- INSERT
 - known restrictions, 2-4
 - operation, 14-2
 - Oracle SQL command, known restrictions, 2-4
 - read-only gateway, 13-6
 - SQL clause, 14-26
 - statement, 14-25, 14-28
- INSERT command
 - known restrictions, 2-4
 - supported by COPY, 13-7
- INSERT command, not supported, 13-6
- installation
 - checklists, 4-3, 5-2, 6-2, 8-2, 11-2, 12-2
 - overview, 4-4
- installing the gateway
 - from CD, 4-4
- internal tracing, C-13
- Internet support, 1-4
- INTERSECT
 - SQL set operators and clauses, 14-21
- IPC
 - adapter, B-4
- ISO standard, 1-5
- isolation level, DRDA_ISOLATION_LEVEL, C-6

J

- JOIN
 - capability, 1-4
- JOIN SQL statement, 14-2

K

keywords

LISTENER, B-4

known problems

compatibility with DB2/UDB, 2-3

known restrictions

accessing DB2 alias objects, 2-4

AVS mapping user IDs, 2-4

bind variables become SQL parameter markers, 2-6

binding the DRDA gateway package on DB2/UDB, 2-5

CONNECT BY not supported, SQL limitations, 2-6

datatype limitations, 2-4

date arithmetic, 2-5

DD basic tables and views, 2-4

dictionary views not provided for DB2/VM, 2-5

DRDA server character sets, 2-4

GLOBAL_NAMES parameter, 2-5

INSERT (Oracle SQL command), 2-4

LONG datatype in SQL*Plus, 2-5

null values and stored procedures, 2-4

Oracle ROWID column, 2-6

row length limitation, 2-5

SAVEPOINT, 2-4

single gateway instances per DRDA network interface, 2-6

SQL*Plus DESCRIBE command, 2-3

string concatenation, 2-5

SUBSTR function post-processed, 2-4

Korn shell

setting ORACLE_HOME variable, 4-4

L

LANGUAGE parameter, D-6

languages, 1-6

SQL*Plus, 1-6

lanscan command, 7-6, 9-6

link station

creating, 7-6, 9-6

parameters, 7-8, 9-8

testing SNA connection, 7-15, 9-14

link, also see Database Link, 14-6

linkage conventions

SIMPLE WITH NULLS, 14-6

linkname, 12-8

listener, B-4

LISTENER keyword, B-4

listener.ora file, 11-5

sample, B-4

literal

character literals, 14-24

date, 14-24

specific datatype, 14-22

TO_DATE, format support, 14-26

log file, gateway installation, 4-6

LOG_DESTINATION parameter, 17-7, 17-8, C-12, C-13

logging, LOG_DESTINATION, C-12

Logical Unit (LU), 6-5, 7-13, 8-5, 9-13

Login as DBA, 4-4

LONG datatype, 14-23

LONG RAW datatype, 14-23

LU

dependent, 7-2, 9-3

additional SNAPPlus2 configuration needed, 7-3, 9-3

Side Information Profiles, 7-3, 9-3

independent vs. dependent, 7-2, 9-2

local LU alias, 7-9

CPI-C destination, 7-12, 9-12

Ethernet parameters, 9-8

local names, 9-8

creating, 7-9, 9-8

names, 7-12, 9-12

partner

creating, 7-9, 9-9

SNAP-IX profiles, 7-2

SNAPPlus2 profiles, 9-2

traffic type, in configuring SNAplus2, 7-7, 9-7

LU name, 15-4

LU6.2

local LU profile, 8-3

profile types, 8-2

side information profile, 8-4

SNA APPC, 6-1, 7-1, 8-1, 9-1

M

MAC address

in SNA configuration, 7-8

in SNAplus2 configuration, 9-8

mapped sqlstate errors, 17-4

mapping user IDs

AVS, 15-4

known restrictions, 2-4

migrating the gateway instance, 16-4

migration problems, 16-2

MINUS

set operator, SQL Set Clauses, 14-21

SQL set operators and clauses, 14-21

mode

name, 7-12, 9-12

profiles, creating, 7-10, 9-10

Mode Profile

SNA Server profile, 8-3

Mode_name

description, 6-4

multi-byte support, D-10

MVS VTAM

configuring DB2/OS390, 5-3

N

National Language Support

initsid.ora parameters, D-5

overview, D-1

Native Semantics

- gateway architecture, 1-12
- parameters, SQL Set Clauses, 14-21
- with SUBSTR function, known restrictions, 2-4
- network
 - attachment, 3-2
 - Oracle Net configuration, 11-4
 - requirements, 3-2
- NLS
 - Also See National Language Support, D-1
 - DRDA server character sets, D-7
- NLS parameters
 - configuration on client and Oracle servers, D-4
- NLS_LANG
 - environmental variable, D-3
 - server-side parameter, D-2
- NLS_LANG environment variable, D-4
- non-character binary data, 14-23
- null
 - rows, 14-27
 - values, 14-27
- number of cursors,
 - DRDA_PACKAGE_SECTIONS, C-9
- numbers
 - concatenation restrictions, 2-5
- numeric datatype, 14-27

O

- o2pc.sql, 12-8, 13-5
- obsolete parameters since V4 gateway, 16-4
- open cursors, at the IBM database, 14-30
- OPEN_LINKS parameter, 13-3
- operating system requirements, 3-3
- operations
 - DELETE, 14-2
 - INSERT, 14-2
 - SELECT, 14-2
 - UPDATE, 14-2
- operators
 - UNION ALL, SQL Set Clauses, 14-21
 - UNION, SQL Set Clauses, 14-21
- option
 - binding packages, 12-6
 - data dictionary views, 14-30
 - date format string, 14-25
 - DRDA session security, 15-4
 - FOR BIT DATA, 14-23
 - Oracle server, 1-8
 - read-only
 - gateway configuration, 1-6
 - replicating, 13-6
 - security conduct, 15-3
 - service port number,
 - DRDA_CONNECT_PARM, C-3
 - SNA security, 6-5, 7-13, 8-5, 9-13
 - SQL functions, 14-19
 - SQL*Plus COPY command, 13-7
 - ORA_MAX_DATE parameter, C-12
 - ORA-NLS33 parameter, C-12
 - description, D-2

- ORA-00001 error, index constraint violated, 17-4
- ORA-00942 error
 - mapped error example, 17-4
 - object name too long, 17-4
- ORA-01017 error, logon denied, 17-4
- ORA-01031 error, insufficient privileges, 17-4
- ORA-01403 error at gateway bind, Bug 3275652, 2-3
- ORA-01460 error, invalid CCSID, 17-4
- ORA-01476 error, divide by zero, 17-4
- ORA-02019 error, 17-2
- ORA-06512 error at gateway bind, Bug 3275652, 2-3
- ORA1 Oracle instance, 14-3
- ORA-1821 with bug 1941672 (fixed), 2-2
- ORA2 Oracle instance, 14-3
- ORA-2025
 - error when using INSERT command, 13-6
- ORA-2068 with Bug 2854129 (fixed), 2-2
- ORA-28500 error
 - was ORA-09100, 17-3
- ORA-28500 with Bug 3121041 (fixed), 2-2
- ORA-28501 error
 - communication error, 17-3
- ORA-28511 with Bug 2854129 (fixed), 2-2
- ORA-28527, conversion errors, C-4
- ORA-9100 to ORA-9199 errors, 17-2
- Oracle
 - directory tree, 4-4
 - error number or return code, 17-2
 - mapped error codes, 17-4
 - products compatibility, 1-10
 - RAW datatype, C-4
 - snapshots, 13-6
- Oracle Database
 - stored procedure, defined, 14-3
- Oracle Database 10g
 - server description, 1-2
- Oracle integrating server
 - accessing gateway from other Oracle Database servers, 12-9
 - accessing other DRDA servers, 12-9
 - accessing the gateway, 13-3
 - architecture, 1-8
 - configuration, 12-8
 - definition, 1-7
 - errors detected, 17-2
 - requirements, 3-4
 - using, in application development, 14-2
- Oracle Net, 1-6, 16-4
 - and application development, 1-14
 - and distributed processing, 11-3
 - and remote data access, 1-13
 - and server coexistence, 1-8
 - API, 11-3
 - compatibility with SQL*Net, 11-3
 - configuring, 11-4
 - distributed
 - database, 11-3
 - processing, 11-3
 - editing to set up security test, 11-6
 - Heterogeneous Services (HS) facility, 11-3

- introduction, 11-3
- operating system authentication, 15-2
- overview, 11-3
- purpose, 1-9
- requirements, 3-4
- sample files, B-1
- sample listener.ora file, B-4
- support for CHECKSUM and encryption, 11-5
- terminology, 11-4
 - client, 11-4
 - driver, 11-4
 - host, 11-4
 - network, 11-4
 - protocol, 11-4
- TNS connect descriptor specification, 13-2
- Oracle ROWID column
 - known restrictions, 2-6
- Oracle Server
 - relationship to host, 1-8
 - services, 1-4
 - database triggers, 1-4
 - distributed capabilities, 1-4
 - distributed query optimization, 1-4
 - SQL, 1-4
 - stored procedures, 1-4
 - two-phase commit protection, 1-4
 - triggers, 13-6
- Oracle server
 - copying data
 - from DRDA server, 13-7
 - to DRDA server, 13-6
- Oracle Universal Installer
 - de-installing the gateway, 4-7
 - starting, 4-6
 - steps to install the gateway, 4-6
 - upgrading for installation, 4-6
- ORACLE_DRDA_TCTL parameter, 17-7, C-12
- ORACLE_DRDA_TRACE parameter, C-13
- ORACLE_HOME
 - environmental variable, 4-4
 - for the gateway, 4-4
- ORACLE2PC table, 12-6, 12-7, 12-8, 13-5
 - DB2/400, 5-4
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
- ORACLE2PC table,
 - DRDA_PACKAGE_OWNER, C-8
- ORADRDA.Oracle2PC table, 13-5
- oraproc1, 14-3
 - stored procedure, 14-3
- oraproc2
 - stored procedure, 14-3
- ORARECOV user ID, 5-6
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
- ORARECOV user ID,
 - DRDA_RECOVERY_USERID, C-9
- ORDER BY clause

- SQL Set Clauses, 14-21
- OS/390 (MVS)
 - configuring DB2/OS390, 5-3

P

- package
 - collection id, DRDA_PACKAGE_COLLID, C-7
 - consistency token,
 - DRDA_PACKAGE_CONSTOKEN, C-8
 - privileges - BIND and EXECUTE, 5-6
 - privileges - BIND, COPY, and EXECUTE, 5-3, 5-7
- packed decimal, 14-27
- parameter
 - changed, 16-3
 - checking settings, 12-7
 - diagnostic, C-12
 - gateway, 5-7
 - link station, 7-8, 9-8
 - Native Semantics, SQL Set Clauses, 14-21
 - new since V4 gateway, 16-3
 - obsolete since V4 gateway, 16-4
 - renamed since V4 gateway, 16-3
 - setting up trace parameters, 17-6
- parameter file
 - tailoring to configure the host, 12-4
- parameters
 - DB_DOMAIN
 - known restrictions, 2-5
 - DRDA_CAPABILITY, 14-19
 - DRDA_CODEPAGE_MAP, C-3, D-7
 - DRDA_DESCRIBE_TABLE
 - known restrictions, 2-4
 - DRDA_DISABLE_CALL, 12-7
 - DRDA_PACKAGE_NAME, 12-7
 - DRDA_PACKAGE_SECTIONS, 14-30
 - DRDA_READ_ONLY, 13-5
 - DRDA_RECOVERY_PASSWORD, 5-6
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DRDA_RECOVERY_USERID, 5-6
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
 - FDS_CLASS, C-10
 - FDS_CLASS_VERSION, C-10
 - FDS_INSTANCE, C-11
- Gateway Initialization File
 - DRDA_CACHE_TABLE_DESC, C-2
 - DRDA_CAPABILITY, C-3
 - DRDA_CMSRC_CM_IMMEDIATE, C-4
 - DRDA_CODEPAGE_MAP, C-3
 - DRDA_COMM_BUFLLEN, C-3
 - DRDA_CONNECT_PARM, 17-3
 - DRDA_CONNECT_PARM (SNA format), C-3
 - DRDA_CONNECT_PARM (TCP/IP
 - format), C-3
 - DRDA_DEFAULT_CCSSID, C-4
 - DRDA_DESCRIBE_TABLE, C-4
 - DRDA_DISABLE_CALL, C-5

- DRDA_FLUSH_CACHE, C-5
- DRDA_GRAPHIC_LIT_CHECK, C-6
- DRDA_GRAPHIC_PAD_SIZE, C-5
- DRDA_GRAPHIC_TO_MBCS, C-6
- DRDA_ISOLATION_LEVEL, C-6
- DRDA_LOCAL_NODE_NAME, C-7
- DRDA_MBCS_TO_GRAPHIC, C-7
- DRDA_OPTIMIZE_QUERY, 13-4, C-7
- DRDA_PACKAGE_COLLID, 17-4, C-7
- DRDA_PACKAGE_CONSTOKEN, C-8
- DRDA_PACKAGE_NAME, 17-4, C-8
- DRDA_PACKAGE_OWNER, C-8
- DRDA_PACKAGE_SECTIONS, C-9
- DRDA_READ_ONLY, C-9
- DRDA_RECOVERY_PASSWORD, C-9
- DRDA_RECOVERY_USERID, C-9
- DRDA_REMOTE_DB_NAME, C-10
- DRDA_SECURITY_TYPE, C-10
- HS_FDS_FETCH_ROWS, C-11
- HS_LANGUAGE, C-11
- HS-NLS_NCHAR, C-11
- LOG_DESTINATION, C-12
- ORA_MAX_DATE, C-12
- ORA-NLS33, C-12
- ORACLE_DRDA_TCTL, C-12
- ORACLE_DRDA_TRACE, C-13
- TRACE_LEVEL, C-13
- HS_DB_DOMAIN
 - known restrictions, 2-5
- HS-NLS_DATE_FORMAT, 14-25, 14-26
- HS_RPC_FETCH_REBLOCKING, 1-11, 14-2
- HS_RPC_FETCH_SIZE, 1-11, 14-2
- LOG_DESTINATION, C-13
- LOG_DESTINATION, 17-7
- OPEN_LINKS, 13-3
- USING, modifying tnsnames.ora file, 11-5
- partner LU
 - creating, 7-9, 9-9
 - creating, also see LU, 7-9
 - see LU
- partner LU alias, 8-3
- partner LU profile, 6-4
- Partner_LU_name
 - description, 6-4
- passthrough, 1-5, 1-11, 14-28, 14-29
 - result sets, 14-28, 14-29
 - SQL feature, 14-28
- patches
 - required for IBM Communications Server for AIX, 3-4
- performance, 14-19
- performance enhancements
 - with fetch reblocking, 14-2
- PL/SQL
 - call, 14-5
 - DRDA stored procedures, 14-4
 - records, 14-6
 - routine, 1-6
 - standard Oracle, 1-6
 - stored procedure, 14-3
- port, 7-6, 9-6
 - adding, for SNA, 7-5, 9-5
 - name, adding, for SNA, 7-6, 9-6
- port number
 - Primary, 5-3
 - DB2/400, 5-4
 - DB2/UDB, 5-5
 - Recovery, 5-3
 - DB2/400, 5-4
 - DB2/UDB, 5-5
- post-processed SQL functions
 - overview, 14-7
- post-processing, 14-19, 17-6
- PREPARE TRANSACTION statement, 13-5
- Primary port number, 5-3
 - DB2/400, 5-4
 - DB2/UDB, 5-5
- privileges
 - BIND
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
 - BINDADD
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
 - BINDAGENT
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
 - CONNECT
 - DB2/UDB, 5-6
 - COPY
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
 - CREATE IN
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
 - CREATE PUBLIC DATABASE LINK, 12-5, 12-8
 - CREATEIN
 - DB2/UDB, 5-6
 - CREATETAB
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - data dictionary emulation, 14-30
 - EXECUTE
 - DB2/OS390, 5-3
 - DB2/UDB, 5-6
 - DB2/VM, 5-7
- procedure
 - stored, 13-6
 - using DRDA server, 14-4
- processing time, with GROUPBY, HAVING, WHERE, 14-21
- product installation directory, 4-4
- profile
 - creating
 - CPI-C and Mode profiles, 7-10, 9-10
 - name
 - see Side Information Profile
 - side information, 7-11, 9-11

- type
 - see Side Information Profile
- profile set, 12-9
- profile types, 8-2
 - Mode Profile, 8-3
- protocol
 - APPC/LU6.2, 6-1, 7-1, 8-1, 9-1
 - commit confirm, 1-6
 - communications protocols, 11-3
 - converter, 1-4
 - definition, 11-4
 - implicit protocol conversion, 1-4
 - network, 13-4
 - Oracle Net Protocol Adapters, 11-3
 - PROTOCOL=IPC, sample, B-3
 - protocol-independent encryption, 1-5
 - two-phase commit, 13-5
- protocols
 - IPC, 11-5
 - SNA, 1-8
 - TCP/IP, 1-4, 1-8, 1-12
 - TCP/IP, gateway transparency, 1-3
- public database link, 12-8

Q

- queries, distributed, 13-4

R

- RAW datatype, 14-23, C-4
- read-only gateway option, 13-5
- read-only gateway option,
 - DRDA_READ_ONLY, C-9
- read-only support, 1-14
- rebind, C-6
 - DRDA_DISABLE_CALL, C-5
 - DRDA_PACKAGE_COLLID, C-7
 - DRDA_PACKAGE_CONSTOKEN, C-8
 - DRDA_PACKAGE_NAME, C-8
 - DRDA_PACKAGE_OWNER, C-8
 - DRDA_PACKAGE_SECTIONS, C-9
- Recovery port number, 5-3
 - DB2/400, 5-4
 - DB2/UDB, 5-5
- recovery user ID and password, 5-3, 5-6
 - DB2/400, 5-5
 - DB2/VM, 5-7
- remote
 - computer, 11-3
 - connections, 13-3
 - data, 1-4
 - data access, 1-13
 - database, 5-6, 12-7, 13-2, 13-6, 17-2, 17-6
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DB2/VM, 5-7
 - database, defining a path, 13-2
 - database, DRDA_PACKAGE_SECTIONS, C-9
 - database, privileges of user/ ID, E-4

- DB2 system, 2-4
- DRDA database,
 - DRDA_ISOLATION_LEVEL, C-6
- DRDA server object information, 2-3
- instance, and Oracle stored procedures, 14-3
- LU, 8-3
- objects, 15-3
- Oracle instance, 14-3, 14-4
- Oracle servers, 11-3
- procedure, 1-6
 - table, 1-5
- transaction program, 3-2, 8-2
- userid and password, 13-2
- Remote Node definition
 - creating, 7-6, 9-6
- renamed parameters, 16-3
- REPLACE
 - command
 - supported by COPY, 13-7
- replication, 13-6
- requirements
 - hardware, 3-2
 - CD-ROM drive, 3-2
 - software, 3-3
- RESULT, 14-5
- result sets, 1-11
 - columns in, 1-12
- return code, 17-2
- REVISE_SALARY
 - stored procedure, 14-5
- ROWID
 - Oracle column
 - known restrictions, 2-6

S

- sample
 - files, B-1
 - listener.ora file, B-4
 - SQL scripts, 12-7
 - tnsnames.ora file, B-3
- samples
 - Side Information File, SunLink SNA
 - Peer-to-Peer, 6-3, 6-5
 - SNA configuration file, 6-3
- SAP
 - dialog box, 7-6, 9-6
 - number, 7-8, 9-8
- SAVEPOINT
 - known restrictions, 2-4
- schema privileges - CREATEIN, 5-6
- scripts
 - dropold.sql, 12-8
 - g4ddtab, 12-8
 - g4ddview.sql, 12-8
- SDLC Coaxial network attachment, 3-2
- security
 - Advanced Security, 1-5
 - DRDA_SECURITY_TYPE, C-10
 - encryption, 11-5

- for session use, 7-12, 9-12
- overview, 15-2
- site autonomy, 1-7
- validation
 - for SNA, 6-5, 7-13, 8-5, 9-13
 - for TCP/IP, 15-3
- SELECT and array size, 1-11
- SELECT operation, 14-2
- SELECT statement, 14-2, 14-7, 14-29
 - read-only gateway, 13-6
- SELECT WHERE
 - SQL clause, 14-26
- semantics, 14-19
- server database error, 17-3
- server profiles, 6-1, 7-1, 8-1, 9-1
- service port number,
 - DRDA_CONNECT_PARM, C-3
- Service TP
 - "07F6C4C2", 7-12, 9-12
- session
 - authorization, 7-12, 9-12
- session, connection, 14-6
- set operators
 - compatibility, SQL Set Clauses, 14-21
 - EXCEPT, SQL Set Clauses, 14-21
 - INTERSECT, SQL Set Clauses, 14-21
 - MINUS, SQL Set Clauses, 14-21
- shift attribute, multi-byte support, D-11
- SID
 - also see Gateway System Identifier, 12-4
 - choosing for gateway, 12-4
 - configuring additional DRDA server
 - instances, 12-9
 - definition, 12-4
- Side Information File
 - information needed, 6-4
 - sample files, 6-3
 - Version 9 sample, 6-5
- Side Information Profile, 7-11, 8-4, 9-11, 17-3
 - defining, to use dependent LUs, 7-3, 9-3
 - description and role, 7-2, 9-2
 - specifies Local LU Profile name, 8-3
- side information profile, 8-4
- SIMPLE linkage convention, 14-6, G-2
- single-threaded conversations for dependent
 - LUs, 7-2, 9-2
- site autonomy, 1-7
- SMIT (System Management Interface Tool)
 - AIX System Management Interface Tool, 8-2
- SNA
 - APPC, 6-1, 8-1, 9-1
 - configuring DB2/400, 5-4
 - configuring DB2/OS390, 5-3
 - configuring DB2/UDB, 5-5
 - configuring DB2/VM, 5-7
 - configuring server profiles, 8-2
 - connectivity via APPC, 7-1
 - conversation security, 6-5, 7-13, 8-5, 9-13
 - CPI-C error, 17-3
 - definitions
 - creating, for SNAP-IX, 7-3
 - creating, for SNAplus2, 9-3
 - creating, for the gateway, 7-3, 9-3
 - maintenance, 7-3, 9-3
 - facilities, 1-12
 - functions, 1-9
 - LU, 8-3
 - network, 7-3, 9-3
 - name, in creating partner LUs, 7-9, 9-9
 - profile, 12-9
 - profile definitions, 8-2
 - profiles, 7-2, 9-2
 - protocol, 1-8, 7-6, 9-6
 - remote access, 1-13
 - sample configuration file, 6-3
 - sample profile definitions, 8-2
 - security
 - option and IBM Communications Server for
 - AIX, 6-6, 7-14, 8-6, 9-14
 - validation, 15-3
 - security validation, 6-5, 7-13, 8-5, 9-13
 - security, DRDA_SECURITY_TYPE, C-10
 - SECURITY=PROGRAM, 6-5, 7-13, 8-5, 9-13
 - SECURITY=SAME, 6-6, 7-14, 8-6, 9-14
 - send/receive buffer, C-3
 - session allocation mode,
 - DRDA_CMSRC_CM_IMMEDIATE, C-4
 - TPN, 8-4
- SNA LU6.2
 - configuring DB2/400, 5-4
 - configuring DB2/OS390, 5-3
- SNA node, 7-2, 7-3, 7-6, 9-2, 9-4, 9-6
 - configuring, 7-5, 9-5
- SNA security option, 6-5, 7-13, 8-5, 9-13
- SNA Server
 - Link Profile, 8-3
 - Local LU Profile, 8-3
 - Partner LU Location Profile, 8-4
 - Partner LU Profile, 8-3
 - profiles
 - creating, 8-2
 - sample profile definitions, 8-2
 - Side Information Profile, 8-4
- SNA server
 - network requirements, 3-2
- sna_domn.cfg file, 7-3, 9-3
- sna_node.cfg file, 7-3, 9-3
- SNAP-IX
 - configuring, 7-1
 - configuring, using xsnaadmin program, 7-2
 - configuring, using xsnapadmin, 7-3
 - definitions
 - for the gateway, 7-3
 - stored in directory, 7-3
 - Side Information Profiles required, 7-2
 - testing the connection, 7-15
- SNAPplus2
 - configuring, 9-1
 - configuring, using xsnapadmin, 9-3
- SNAPplus2

- additional configuration needed for multiple dependent LUs, 7-3, 9-3
- configuring, using xsnapadmin program, 9-2
- definitions
 - for the gateway, 9-3
 - stored in directory, 9-3
- Side Information Profiles required, 9-2
- testing the connection, 9-14
- version required, 3-4
- snapshots
 - known restrictions, SQL limitations, 2-6
 - Oracle Snapshot feature, 13-6
- software requirements, 3-3
- Solaris
 - system support for APPC, 6-1
- sort order
 - with ORDERBY, 14-21
- SPUFI, a database native tool, 12-7
- SQL
 - ANSI standard, 1-5
 - clause compatibility, 14-21
 - clauses
 - DELETE, 14-26
 - INSERT, 14-26
 - SELECT WHERE, 14-26
 - UPDATE, 14-26
 - commands
 - DESCRIBE, known restrictions, 2-3
 - constructs
 - Oracle processing, 14-6
 - differences in the gateway, 1-10
 - errors mapped to Oracle error codes, 17-4
 - functions
 - SUBSTR, 14-19
 - functions and Native Semantics, 14-19
 - functions list, F-1
 - gateway architecture, 1-3
 - gateway transparency, 1-5
 - ISO standard, 1-5
 - native DB2, 1-5
 - passthrough, 14-28, 14-29
 - statements, 14-2
 - DB2, 14-28
 - issued through the gateway, 13-7
 - passing through gateway, 14-28
 - statements, DRDA_ISOLATION_LEVEL, C-6
 - syntax, 14-27
 - tracing, not to be used in production environment, 13-7
- SQL functions
 - column functions, 14-7
 - compatible, defined, 14-7
 - compensated, defined, 14-7
 - DB2/400, 14-13
 - DB2/OS390, 14-8
 - DB2/UDB, 14-10
 - DB2/VM, 14-16
 - post-processing, defined, 14-7
 - that can be disabled, 14-21
 - that can be enabled, 14-19
 - translated, defined, 14-7
 - with Native Semantics, 14-19
- SQL tracing
 - in Oracle Database, 17-6
 - in the gateway, 17-7
 - LOG_DESTINATION, 17-7
 - ORACLE_DRDA_TCTL parameter, 17-7
 - TRACE_LEVEL parameter, 17-7
- SQL*Net
 - gateway migration problems, 16-2
 - Heterogeneous Services (HS) facility, 11-3
 - replaced by Oracle Net, 11-3
- SQL*Plus
 - connecting to gateway, 11-6
 - COPY command, 13-7
 - DESCRIBE command, known restrictions, 2-3
 - extending gateway uses, 1-11
 - moving data, 1-6
- sqlstate, mapped errors, 17-4
- stability, of cursor,
 - DRDA_ISOLATION_LEVEL, C-6
- Startup Shell Script
 - parameter, tailoring to configure the host, 12-4
- statement
 - CREATE DATABASE, client/server configuration, D-4
- statements
 - CREATE DATABASE LINK, 13-2
 - DB2 CREATE TABLE, 1-5
 - DDL, 14-28
 - DELETE, 14-25, 14-28
 - DROP DATABASE LINK, 13-2
 - GRANT, 12-9
 - INSERT, 14-28
 - PREPARE TRANSACTION, 13-5
 - SELECT, 14-2, 14-7, 14-29
 - read-only gateway, 13-6
- SQL
 - DB2, 14-28
 - JOIN, 14-2
 - SELECT, 14-29
 - UPDATE, 14-25, 14-28
- steps for
 - configuring communications interfaces, 7-2, 9-2
- stored definitions
 - Side Information Profile needed for SNAP-IX, 7-2
 - Side Information Profile needed for
 - SNAPplus2, 9-2
- stored procedure
 - creating on DB2, 14-5
 - DB2, 14-6
 - native DB2, 1-6
 - Oracle and non-Oracle, 1-5
 - Oracle Database Server
 - local instance, 14-3
 - Oracle Database server
 - PL/SQL, 14-3
 - remote instance, 14-3
 - Oracle Server
 - using, 14-3

- Oracle, description, 1-6
- restriction, 2-4
- usage, C-5
- using DRDA server, 14-4
- stored procedures, 1-4
 - DB2, 13-6, 14-4
 - GTW\$_BIND_PKG, 12-5
 - REVISE_SALARY, 14-5
 - using with the gateway, 14-3
- string concatenation
 - known restrictions, 2-5
- string index, with Native Semantics, 14-19
- Structured Query Language, also see SQL, 1-3
- SUBSTR SQL function, 14-19
 - known restrictions, 2-4
 - with Native Semantics, known restrictions, 2-4
- Sun host
 - configuring the SunLink SNA Peer-to-Peer, 6-3
 - defining the gateway name, 6-3
- SunLink SNA Peer-to-Peer product
 - configuring
 - side information record file, 6-4
 - sample configuration files shipped, 6-3
 - setting up a configuration file, 6-3
 - setting up a gateway name, 6-3
 - starting Version 9, 6-4
- synonym, 1-10
 - feature, 13-4
 - for location transparency, 14-3
- system privileges - BINDADD and BINDAGENT, 5-3, 5-7

T

- table
 - create a table in DB2, 14-29
 - insert a row into a DB2 table, 14-29
- TABLE_PRIVILEGES view, A-12
- tables
 - EXPLAIN_PLAN, 1-12
 - ORACLE2PC, 12-7, 12-8, 13-5
 - ORADRD.A.ORACLE2PC, 13-5
- TCP/IP
 - adapter, B-4
 - concurrent connections, 3-2
 - configuring
 - DB2/400, 5-4
 - DB2/OS390, 5-3
 - DB2/UDB, 5-5
 - DB2/VM, 5-7
 - configuring under Solaris, 10-2
 - DRDA_CONNECT_PARM, C-3
 - facilities, 1-12
 - format, gateway initialization file parameter, C-3
 - functions, 1-9
 - modifying tnsnames.ora in Oracle Net, 11-5
 - network attachment, 3-2
 - number of database link connections, 14-6
 - protocol, 1-4, 1-8
 - protocol, gateway transparency, 1-3

- remote access, 1-13
- security, 15-3
- security validation, 15-3
- support, 1-12
- terminology defined, 1-7
- tg4drda directory, also known as
 - ORACLE_HOME, 4-4
- TIME datatype, 14-24
- time operations, 14-24
- TIMESTAMP datatype, 14-24
- TNSNAMES.ORA
 - changes to, during migration problems, 16-2
- tnsnames.ora
 - connect descriptor, 13-2
 - file, 11-5, 12-8, 12-9
 - modifying for Oracle Net, 16-4
 - sample file, B-3
- TO_DATE
 - function, 14-24, 14-25, 14-26
- TO_DATE function
 - main topic, 14-26
- Token Ring, 3-2
- token, package consistency,
 - DRDA_PACKAGE_CONSTOKEN, C-8
- tools and the gateway, 1-10
- TP name, 7-12, 9-12
- TP_name
 - description, 6-4
- TPN, 8-4
- trace control, C-12
- trace parameters
 - setting, 17-6
- TRACE_LEVEL parameter, 17-7, C-13
- tracing
 - code, C-13
 - ORACLE_DRDA_TRACE, C-13
 - SQL statements, 13-7
- tracing, LOG_DESTINATION, C-12
- trade-off, Native Semantics, 14-19
- transaction mode, read-only,
 - DRDA_READ_ONLY, C-9
- transaction program name, remote, 8-4
- transactions
 - CICS, 14-6
 - IMS, 14-6
- transform, character set transforms with multi-byte support, D-10
- transform, not required for DRDA Server, D-11
- transparency
 - main topic, gateway transparency, 1-3
 - native semantics, 14-19
- triggers
 - for Oracle Server, 13-6
- TSO
 - configuring DB2/OS390, 5-3
- two-phase commit
 - ORACLE2PC table
 - DB2/400, 5-4
 - DB2/UDB, 5-6
 - DB2/VM, 5-7

- ORACLE2PC table, DB2/OS390, 5-3
- processing transactions, 13-5
- protection, 1-4
- unsupported statement, 13-5

U

UNION

- capability, 1-4
- operator, SQL Set Clauses, 14-21
- SQL set operators and clauses, 14-21

UNION ALL

- operator, SQL Set Clauses, 14-21
- SQL set operators and clauses, 14-21

UPDATE

- known restrictions, SQL limitations, 2-6
- operation, 14-2
- read-only gateway, 13-6
- SQL clause, 14-26
- statement, 14-28

user ID mapping

- AVS, 15-5
- DB2/400, 15-5
- DB2/OS390, 15-4
- DB2/VM, 15-4

user ID translation, DB2, 15-5

user privileges, 14-30

USER_CATALOG view, A-12

USER_COL_COMMENTS view, A-12

USER_CONS_COLUMNS view, A-13

USER_CONSTRAINTS view, A-13

USER_DB_LINKS data dictionary view, 13-3

USER_INDEXES view, A-14

USER_OBJECTS view, A-15

USER_SYNONYMS view, A-16

USER_TAB_COLUMNS view, A-18

USER_TAB_COMMENTS view, A-19

USER_TABLES view, A-16

USER_USERS view, A-19

USER_VIEWS view, A-19

USING clause, 13-2

USING parameter, modifying tnsnames.ora file, 11-5

Using the gateway, 13-1

V

VALUES clause

- functions not allowed by DB2, 14-26

VARCHAR datatype, 14-23

variable

- bind, SQL passthrough, 14-28
- input bind, 14-25

view

- creating, 12-9
- data dictionary
 - emulation on DRDA server, 14-30

views

- catalog
 - DB2/400, 14-30

- DB2/OS390, 14-30

- DB2/UDB, 14-30

- DB2/VM, 14-30

VM VTAM

- configuring, 5-7

VSAM

- file, 14-6

VTAM

- configuring DB2/OS390, 5-3

definitions

- relationship to SNAplus2 definitions, 7-15, 9-14

- same local LU name as on remote DRDA server, 7-9, 9-8

- security validation, 6-5, 7-13, 8-5, 9-13

W

WHERE clause

- known restrictions, SQL limitations, 2-6
- SQL Set Clauses, 14-21

WHERE CURRENT OF CURSOR clause

- known restrictions, SQL limitations, 2-6

X

xснаadmin

- creating SNA definitions, 7-3

- invoking, 7-3

- SNAP-IX definitions, 7-3

- tool to configure SNAP-IX, 7-2

- tool to create SNAP-IX definitions, 7-3

xsnapadmin

- creating SNA definitions, 9-3

- invoking, 9-4

- tool to configure SNAplus2, 9-2

- tool to create SNAplus2 definitions, 9-3

X-Windows, 7-2, 9-2

Z

zoned decimal operations, 14-27