## Contents

Send Us Your Comments ........................................................................................................... xvii

Preface ............................................................................................................................................ xix

1 Introduction

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to the Oracle Transparent Gateway</td>
<td>1-1</td>
</tr>
<tr>
<td>Protection of Current Investment</td>
<td>1-1</td>
</tr>
<tr>
<td>Release 9i Gateways</td>
<td>1-3</td>
</tr>
<tr>
<td>Advantages of the Gateway</td>
<td>1-3</td>
</tr>
<tr>
<td>Gateway Capabilities</td>
<td>1-4</td>
</tr>
<tr>
<td>Transparency at All Levels</td>
<td>1-4</td>
</tr>
<tr>
<td>Extended Database Services</td>
<td>1-5</td>
</tr>
<tr>
<td>Extended Advanced Networking, Internet and Intranet Support</td>
<td>1-6</td>
</tr>
<tr>
<td>Dynamic Dictionary Mapping</td>
<td>1-6</td>
</tr>
<tr>
<td>SQL</td>
<td>1-7</td>
</tr>
<tr>
<td>Data Definition Language</td>
<td>1-7</td>
</tr>
<tr>
<td>Data Control Language</td>
<td>1-7</td>
</tr>
<tr>
<td>Passthrough and Native DB2 SQL</td>
<td>1-7</td>
</tr>
<tr>
<td>Stored Procedures</td>
<td>1-8</td>
</tr>
<tr>
<td>Languages</td>
<td>1-8</td>
</tr>
<tr>
<td>Oracle Server Technology and Tools</td>
<td>1-8</td>
</tr>
<tr>
<td>SQL*Plus</td>
<td>1-9</td>
</tr>
<tr>
<td>Two-Phase Commit and Multisite Transactions</td>
<td>1-9</td>
</tr>
<tr>
<td>Site Autonomy</td>
<td>1-9</td>
</tr>
<tr>
<td>Migration and Coexistence</td>
<td>1-9</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Security</td>
<td>1-10</td>
</tr>
<tr>
<td>Terms</td>
<td>1-10</td>
</tr>
<tr>
<td>Architecture</td>
<td>1-11</td>
</tr>
<tr>
<td>Implementation</td>
<td>1-12</td>
</tr>
<tr>
<td>How the Gateway Works</td>
<td>1-13</td>
</tr>
<tr>
<td>SQL Differences</td>
<td>1-14</td>
</tr>
<tr>
<td>Oracle Tools and the Gateway</td>
<td>1-14</td>
</tr>
<tr>
<td>SQL*Plus</td>
<td>1-14</td>
</tr>
<tr>
<td>Features</td>
<td>1-15</td>
</tr>
<tr>
<td>Heterogeneous Services Architecture</td>
<td>1-15</td>
</tr>
<tr>
<td>Performance Enhancements</td>
<td>1-15</td>
</tr>
<tr>
<td>Fetch Reblocking</td>
<td>1-15</td>
</tr>
<tr>
<td>Oracle 9i Passthrough Supported</td>
<td>1-15</td>
</tr>
<tr>
<td>Retrieving Result Sets Through Passthrough</td>
<td>1-16</td>
</tr>
<tr>
<td>Support for TCP/IP</td>
<td>1-16</td>
</tr>
<tr>
<td>Native Semantics</td>
<td>1-16</td>
</tr>
<tr>
<td>Columns Supported in a Result Set</td>
<td>1-16</td>
</tr>
<tr>
<td>EXPLAIN_PLAN Improvement</td>
<td>1-16</td>
</tr>
<tr>
<td>Heterogeneous Database Integration</td>
<td>1-16</td>
</tr>
<tr>
<td>Minimum Impact on Existing Systems</td>
<td>1-17</td>
</tr>
<tr>
<td>Large Base of Data Access</td>
<td>1-17</td>
</tr>
<tr>
<td>Application Portability</td>
<td>1-17</td>
</tr>
<tr>
<td>Remote Data Access</td>
<td>1-18</td>
</tr>
<tr>
<td>Support for Distributed Applications</td>
<td>1-19</td>
</tr>
<tr>
<td>Application Development and End User Tools</td>
<td>1-19</td>
</tr>
</tbody>
</table>

## 2 Release Information

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Set</td>
<td>2-2</td>
</tr>
<tr>
<td>Changes and Enhancements, Release 9.2.0.1.0</td>
<td>2-2</td>
</tr>
<tr>
<td>Enhancements in Release 9.0.1</td>
<td>2-2</td>
</tr>
<tr>
<td>Changes in Release 9.0.1</td>
<td>2-3</td>
</tr>
<tr>
<td>Known Problems</td>
<td>2-5</td>
</tr>
<tr>
<td>Known Restrictions</td>
<td>2-6</td>
</tr>
<tr>
<td>DB2 Considerations</td>
<td>2-6</td>
</tr>
<tr>
<td>SQL Limitations</td>
<td>2-9</td>
</tr>
</tbody>
</table>
3 System Requirements

Hardware Requirements ................................................................. 3-2
  Processor ............................................................... 3-2
  Memory ............................................................... 3-2
  Network Attachment ......................................................... 3-2
  CD-ROM Drive ............................................................ 3-2
  Disk Space ............................................................... 3-3

Software Requirements ............................................................. 3-3
  Operating System .......................................................... 3-3
  DRDA Databases ........................................................... 3-3
  Communication Server ....................................................... 3-4
  Oracle Server .............................................................. 3-4
  Oracle Networking Products ............................................... 3-4

Documentation Requirements ...................................................... 3-5

4 Installing the Gateway

Introduction ...................................................................................... 4-2

Before You Begin ............................................................................. 4-2

Checklist for Gateway Installation .................................................... 4-3

Installation Overview ........................................................................ 4-4

Before Beginning Installation .......................................................... 4-4

Installing the Gateway from CD-ROM ............................................. 4-4
  Step 1: Log on to the host ......................................................... 4-4
  Step 2: Create the product installation directory .................. 4-5
  Step 3: Set the ORACLE_HOME environment variable .......... 4-5
  Step 4: Mount the CD-ROM ............................................... 4-5
  Step 5: Set the DISPLAY Variable ...................................... 4-6
  Step 6: Start the Oracle Universal Installer ....................... 4-6
  Step 7: Step through the Oracle Universal Installer .......... 4-7
  Step 8: Verify installation success ...................................... 4-7

Installation Complete ........................................................................ 4-8

De-installing the Gateway ............................................................... 4-8
5 Configuring the DRDA Server

Checklists for Configuring the DRDA Server ................................................................. 5-2
  DB2/OS390 ..................................................................................................................... 5-2
  DB2/400 ......................................................................................................................... 5-2
  DB2/UDB (Universal Database) ..................................................................................... 5-2
  DB2/VM ......................................................................................................................... 5-2
  DB2/OS390 ..................................................................................................................... 5-3
  Step 1: Configure the Communications Server ............................................................. 5-3
  Step 2: Define the user ID that owns the package ......................................................... 5-3
  Step 3: Define the recovery user ID ............................................................................. 5-4
  Step 4: Determine DRDA location name for DB2 instance ........................................... 5-4
  Step 5: Configure DB2 Distributed Data Facility for gateway ...................................... 5-4
  DB2/400 ......................................................................................................................... 5-5
  Step 1: Configure the Communications Server ............................................................. 5-5
  Step 2: Define the user ID that owns the package ......................................................... 5-5
  Step 3: Define the recovery user ID ............................................................................. 5-5
  Step 4: Determine DRDA location name for DB2/400 instance .................................... 5-6
  DB2/UDB (Universal Database) ..................................................................................... 5-7
  Step 1: Configure the SNA Communications Server ..................................................... 5-7
  Step 2: Define the user ID that owns the package ......................................................... 5-7
  Step 3: Define the recovery user ID ............................................................................. 5-8
  Step 4: Determine DRDA location name for DB2/UDB instance .................................... 5-8
  DB2/VM ......................................................................................................................... 5-9
  Step 1: Configure the Communications Server ............................................................. 5-9
  Step 2: Define the user ID that owns the package ......................................................... 5-9
  Step 3: Define the recovery user ID ............................................................................. 5-9
  Step 4: Determine DRDA location name for DB2/VM instance .................................... 5-10

6 Configuring SunLink for the Gateway

Checklist for Configuring the Communications Interfaces ............................................. 6-2
  Step 1: Setting up a Gateway Name .............................................................................. 6-3
  Step 2: Setting up a Configuration File ......................................................................... 6-3
  Starting the SunLink Peer-to-Peer Version 9 Software ................................................ 6-4
  Step 3: Side Information File ....................................................................................... 6-4
  Partner_LU_name ........................................................................................................ 6-5
7 Configuring SNAP-IX Interfaces

Steps for Configuring the Communications Interfaces .......................................................... 7-2
Before You Begin...................................................................................................................... 7-2
SNAP-IX Configuration Tool ................................................................................................. 7-2
Creating SNAP-IX Profiles for the Gateway ......................................................................... 7-2
Independent Versus Dependent LUs ..................................................................................... 7-2
Creating SNA Definitions for the Gateway ............................................................................ 7-3
  Sample SNAP-IX Definitions .................................................................................................. 7-4
  Configuring SNAP-IX ........................................................................................................... 7-4
  Invoking xsnaadmin ............................................................................................................... 7-4
Using SNA Session Security Validation ................................................................................ 7-17
SNA Conversation Security .................................................................................................. 7-17
  SNA Security Option SECURITY=PROGRAM ....................................................................... 7-17
  SNA Security Option SECURITY=SAME .............................................................................. 7-18
Testing the Connection .......................................................................................................... 7-19

8 Configuring IBM Communication Server

Checklist for Configuring the Communications Interfaces .................................................... 8-2
Step 1: Configuring Communication Server Profiles ............................................................. 8-3
Step 2: Creating Communication Server Profiles for the Gateway ..................................... 8-3
  Sample Profile Definitions .................................................................................................... 8-3
  Profile Types....................................................................................................................... 8-3
Step 3: Testing the Connection .............................................................................................. 8-6
Using SNA Session Security Validation ............................................................................... 8-7
SNA Conversation Security .................................................................................................. 8-8
  SNA Security Option SECURITY=PROGRAM ....................................................................... 8-8
9 Configuring SNAPlus2

Steps for Configuring the Communications Interfaces .............................................. 9-2
Before You Begin ........................................................................................................ 9-2
SNAPlus2 Configuration Tool ..................................................................................... 9-2
Creating SNAPlus2 Profiles for the Gateway ............................................................. 9-2
Independent Versus Dependent LUs .......................................................................... 9-2
Creating SNA Definitions for the Gateway ................................................................. 9-3
   Sample SNAPlus2 Definitions .................................................................................. 9-4
   Configuring SNAPlus2 ............................................................................................ 9-4
   Invoking xsnapadmin ............................................................................................... 9-4
Using SNA Session Security Validation .................................................................... 9-18
SNA Conversation Security .......................................................................................... 9-19
   SNA Security Option SECURITY=PROGRAM ......................................................... 9-19
   SNA Security Option SECURITY=SAME ................................................................. 9-20
Testing the Connection ............................................................................................... 9-21

10 Configuring TCP/IP

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

11 Configuring the Gateway

Configuration Checklists ............................................................................................ 11-2
Choosing a Gateway System Identifier (SID) ........................................................... 11-4
Gateway Configuration and the Startup Shell Script ............................................... 11-4
Configuring the Host .................................................................................................. 11-4
   Step 1: Choose an appropriate Startup Shell Script ................................................ 11-4
   Step 2: Create the gateway Startup Shell Script ...................................................... 11-5
   Step 3: Tailor the Startup Shell Script .................................................................... 11-6
   Step 4: Choose the init$SID.ora file .................................................................... 11-6
   Step 5: Tailor the init$SID.ora file ........................................................................ 11-7
Binding the DRDA Gateway Package ......................................................................... 11-8
Binding Packages on DB2/Universal Database (DB2/UDB) ......................................... 11-9
Dropping Database Links ................................................................. 13-3
Examining Available Database Links ........................................... 13-3
Limiting the Number of Active Database Links ......................... 13-3
Accessing the Gateway ................................................................. 13-4
  Step 1: Login to the Oracle Integrating Server ......................... 13-4
  Step 2: Create a database link to the DRDA database .......... 13-4
  Step 3: Retrieve data from the DRDA database ................. 13-4
Accessing AS/400 File Members .................................................. 13-5
Using the Synonym Feature ......................................................... 13-5
Performing Distributed Queries .................................................. 13-6
  Example of a Distributed Query .............................................. 13-7
  Two-Phase Commit Processing ............................................. 13-8
  Distributed DRDA Transactions ............................................ 13-8
Read-Only Gateway ................................................................. 13-9
Replicating in a Heterogeneous Environment ............................. 13-9
  Oracle9i Server Triggers ....................................................... 13-9
  Oracle Snapshots ............................................................... 13-9
Copying Data from the Oracle9i Server to the DRDA Server ....... 13-10
Copying Data from the DRDA Server to the Oracle9i Server ....... 13-10
Tracing SQL Statements ............................................................ 13-11

14  Developing Applications

Gateway Appearance to Application Programs ............................ 14-2
  Fetch Reblocking ............................................................... 14-3
Using Oracle Stored Procedures with the Gateway ...................... 14-3
Using DRDA Server Stored Procedures with the Gateway .......... 14-5
  Oracle Application and DRDA Server Stored Procedure Completion ......................... 14-6
  Procedural Feature Considerations with DB2 .......................... 14-7
Database Link Behavior ......................................................... 14-7
Oracle Server SQL Construct Processing .................................. 14-8
  Compatible SQL Functions ................................................. 14-8
  Translated SQL Functions ................................................. 14-8
  Compensated SQL Functions ............................................. 14-9
  Native Semantic SQL Functions ......................................... 14-9
  DB2/OS390 SQL Compatibility ........................................... 14-10
15-8
Using the g4drpwd Utility ................................................................. 15-9

16 Migration and Coexistence with Existing Gateways

Migrating Existing V4 Gateway Instances to New Release................................. 16-2
  Step 1: Install the new Release ........................................................................ 16-2
  Step 2: Transferring initsid.gtwboot Gateway Boot Initialization parameters. ........ 16-2
  Step 3: Transferring initsid.ora Gateway Initialization File parameters. .............. 16-2
Backout Considerations When Migrating to New Releases ................................ 16-3
New and Changed Parameters When Migrating from Version 4 to Release 9i of the Gateway ................................................................. 16-3
  New Parameters ............................................................................................... 16-3
  Parameters That Have Been Changed in Usage ................................................. 16-4
  Parameters That Have Been Renamed ............................................................... 16-5
  Obsolete Parameters ....................................................................................... 16-5
DRDA Server Considerations .............................................................................. 16-6
Oracle Net Considerations ................................................................................. 16-6

17 Error Messages, Diagnosis, and Reporting

Interpreting Gateway Error Messages ............................................................... 17-2
  Errors Detected by the Oracle Integrating Server .............................................. 17-2
  Errors Detected by the Gateway ....................................................................... 17-3
  Errors Detected in the DRDA Software ............................................................ 17-3
  Communication Errors ..................................................................................... 17-4
  Errors Detected by the Server Database ............................................................ 17-4
Mapped Errors .................................................................................................... 17-5
Gateway Error Codes .......................................................................................... 17-6
SQL Tracing and the Gateway ............................................................................. 17-8
  SQL Tracing in the Oracle Database ................................................................. 17-8
  SQL Tracing in the Gateway ............................................................................. 17-8

A Oracle DB2 Data Dictionary Views

  Supported Views ............................................................................................... A-2
  Data Dictionary View Tables ............................................................................ A-4
B Sample Files

Sample Gateway Initialization File ................................................................. B-2
Sample Startup Shell Script ................................................................. B-3
Sample Oracle Net tnsnames.ora File ................................................................. B-3
Sample Oracle Net listener.ora File ................................................................. B-4
## C  DRDA-Specific Parameters

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modifying the Gateway Initialization File</td>
<td>C-2</td>
</tr>
<tr>
<td>Setting Parameters in the Gateway Initialization File</td>
<td>C-2</td>
</tr>
<tr>
<td>Syntax and Usage</td>
<td>C-2</td>
</tr>
<tr>
<td>Gateway Initialization File Parameters</td>
<td></td>
</tr>
<tr>
<td>DRDA_CAPABILITY</td>
<td>C-3</td>
</tr>
<tr>
<td>DRDA_CODEPAGE_MAP</td>
<td>C-3</td>
</tr>
<tr>
<td>DRDA_COMM_BUFLEN</td>
<td>C-3</td>
</tr>
<tr>
<td>DRDA_CONNECT_PARM (SNA format)</td>
<td>C-4</td>
</tr>
<tr>
<td>DRDA_CONNECT_PARM (TCP/IP format)</td>
<td>C-4</td>
</tr>
<tr>
<td>DRDA_CMSRC_CM_IMMEDIATE</td>
<td>C-4</td>
</tr>
<tr>
<td>DRDA_DEFAULT_CCSID</td>
<td>C-5</td>
</tr>
<tr>
<td>DRDA_DESCRIBE_TABLE</td>
<td>C-5</td>
</tr>
<tr>
<td>DRDA_DISABLE_CALL</td>
<td>C-6</td>
</tr>
<tr>
<td>DRDA_FLUSH_CACHE</td>
<td>C-6</td>
</tr>
<tr>
<td>DRDA_ISOLATION_LEVEL</td>
<td>C-7</td>
</tr>
<tr>
<td>DRDA_LOCAL_NODE_NAME</td>
<td>C-7</td>
</tr>
<tr>
<td>DRDA_OPTIMIZE_QUERY</td>
<td>C-8</td>
</tr>
<tr>
<td>DRDA_OVERRIDE_FROM_CODEPAGE (obsolete)</td>
<td>C-8</td>
</tr>
<tr>
<td>DRDA_OVERRIDE_TO_CODEPAGE (obsolete)</td>
<td>C-9</td>
</tr>
<tr>
<td>DRDA_PACKAGE_COLLID</td>
<td>C-9</td>
</tr>
<tr>
<td>DRDA_PACKAGE_CONSTOKEN</td>
<td>C-10</td>
</tr>
<tr>
<td>DRDA_PACKAGE_NAME</td>
<td>C-10</td>
</tr>
<tr>
<td>DRDA_PACKAGE_OWNER</td>
<td>C-11</td>
</tr>
<tr>
<td>DRDA_PACKAGE_SECTIONS</td>
<td>C-11</td>
</tr>
<tr>
<td>DRDA_RDBMS_TYPE (obsolete)</td>
<td>C-11</td>
</tr>
<tr>
<td>DRDA_READ_ONLY</td>
<td>C-12</td>
</tr>
<tr>
<td>DRDA_RECOVERY_PASSWORD</td>
<td>C-12</td>
</tr>
<tr>
<td>DRDA_RECOVERY_USERID</td>
<td>C-12</td>
</tr>
<tr>
<td>DRDA_REMOTE_DB_NAME</td>
<td>C-13</td>
</tr>
<tr>
<td>DRDA_SECURITY_TYPE</td>
<td>C-13</td>
</tr>
<tr>
<td>FDS_CLASS</td>
<td>C-14</td>
</tr>
<tr>
<td>FDS_CLASS_VERSION</td>
<td>C-14</td>
</tr>
<tr>
<td>FDS_INSTANCE</td>
<td>C-15</td>
</tr>
<tr>
<td>HS_FDS_FETCH_ROWS</td>
<td>C-15</td>
</tr>
</tbody>
</table>
D National Language Support

Overview of NLS Interactions ................................................................. D-2
Client and Oracle Integrating Server Configuration .......................... D-4
Gateway Language Interaction with DRDA Server ......................... D-5
  Gateway Configuration ................................................................. D-6
  NLS Parameters in the Gateway Initialization File ....................... D-6
  NLS Parameters in the Gateway Startup Shell Script ................... D-8
Gateway Codepage Map Facility ......................................................... D-9
Message Availability ........................................................................ D-11
Example of NLS Configuration ......................................................... D-12

E Configuration Worksheet

F Quick Reference to Oracle SQL Functions

G Sample Applications

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

Index
Oracle Corporation 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.
The Oracle Transparent Gateway for IBM 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 IBM 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 AIX operating system before using this guide to install or administer the gateway.
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 Corporation 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.

Related Publications

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

- Oracle9i Heterogeneous Connectivity Administrator's Guide
  This guide contains information common to all heterogeneous gateways, including important information on functions, parameters, and error messages.
**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 table. The first column lists the conventions used in this manual and the second column describes their meanings:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>example text</code></td>
<td>Words or phrases, such as <code>mkdir</code> and <code>ORACLE</code>, must be entered exactly as spelled and in the letter case shown. In this example, <code>mkdir</code> must be entered in lowercase letters and <code>ORACLE</code> in uppercase letters.</td>
</tr>
<tr>
<td><code>italic text</code></td>
<td>Italicized uppercase or lowercase, such as <code>your_name</code>, indicates that you must substitute a word or phrase, such as the actual directory name.</td>
</tr>
<tr>
<td><strong><code>BOLD text</code> or <code>bold italic TEXT</code></strong></td>
<td><strong>Bold</strong> words or phrases refer to a file directory structure, such as a directory, path or file ID.</td>
</tr>
<tr>
<td><code>{ }</code></td>
<td>Curly braces indicate that one of the enclosed arguments is required. Do not enter the braces themselves.</td>
</tr>
<tr>
<td><code>[ ]</code></td>
<td>Brackets enclose optional clauses from which you can choose one or none. Do not enter the brackets themselves.</td>
</tr>
<tr>
<td><code>...</code></td>
<td>Ellipses indicate that the preceding item can be repeated. You can enter an arbitrary number of similar items.</td>
</tr>
<tr>
<td>`</td>
<td>`</td>
</tr>
</tbody>
</table>

Other punctuation, such as commas, quotes, or the pipe symbol (`|`), must be entered as shown unless otherwise specified. Directory names, file IDs, and so on, appear in the required letter case in examples. The same convention is used when these names appear in text, and the names are highlighted in **bold**. The use of *italics* indicates that those portions of a file ID that appear in *italics* can vary.

Gateway commands, file IDs, reserved words, and keywords appear in uppercase in examples and text. UNIX commands, environment variables, and keywords
appear in the required letter case in examples and text. Reserved words and keywords must always be entered as shown. They have reserved meanings within the Oracle system.

**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.

**UNIX Prompts**

The UNIX prompt, `$`, appears in UNIX command examples. Enter your response at the prompt. Do not enter the text of the prompt, `$`, in your response. A dollar sign ($) is part of some UNIX directory names and should not be confused as a prompt character.

**Storage Measurements**

Storage measurements use these abbreviations:

- KB, for kilobyte, which equals 1024 bytes
- MB, for megabyte, which equals 1,048,576 bytes
- GB, for gigabyte, which equals 1,073,741,824 bytes
Documents Referenced in this Guide

**Oracle9i Books:**

- Oracle9i Administrator’s Reference
- Oracle Advanced Security Administrator’s Guide
- Oracle9i Application Developer’s Guide - Fundamentals
- Oracle C++ Call Interface Programmer’s Guide
- Oracle Call Interface Programmer’s Guide
- Oracle9i Database Administrator’s Guide
- Oracle9i Database Error Messages
- Oracle9i Database Reference
- Oracle9i Database Utilities
- Oracle9i Heterogeneous Connectivity Administrator’s Guide
- Oracle9i Net Services Administrator’s Guide
- Oracle9i Net Services Reference Guide
- Oracle SNMP Support Reference Guide
- Oracle9i SQL Reference
- PL/SQL User’s Guide and Reference
- SQL*Plus User’s Guide and Reference
- Oracle Transparent Gateway for IBM DRDA Installation and User’s Guide
- Oracle Universal Installer Concepts Guide

**Oracle9i Books:**

Oracle9i Administrator’s Reference Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris

Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris

**IBM Books:**

Refer to the IBM documents for your platform and operating system.
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.

Introduction to the Oracle Transparent Gateway

Company growth and diversification often means 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 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.

Protection of Current Investment

Oracle Transparent Gateway for IBM 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 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.
The Oracle Transparent Gateway for IBM 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 server data.
- Read this chapter for information about the architecture, uses, and features of the Oracle Transparent Gateway for IBM DRDA. It contains the following sections:
  - Release 9i Gateways on page 1-3
  - Gateway Capabilities on page 1-4
  - Terms on page 1-10
  - Architecture on page 1-11
  - Implementation on page 1-12
  - How the Gateway Works on page 1-13
  - Oracle Tools and the Gateway on page 1-14
  - Features on page 1-15
Release 9i Gateways

The Oracle9i server provides the foundation for the next generation of the Oracle Open Gateways Version 9, which will deliver enhanced integration capabilities by exploiting Oracle9i Heterogeneous Services. Heterogeneous Services is a component of the Oracle9i server. The Oracle9i server provides the common architecture for future generations of the gateways. For detailed information on Oracle Heterogeneous Services, refer to Oracle9i Heterogeneous Connectivity Administrator’s Guide.

The version 9 gateways are even more tightly integrated with the Oracle9i 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 Oracle9i 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 9 gateways leverage any enhancements in the Oracle9i server, and you can quickly extend those benefits to your non-Oracle data.

Advantages of the Gateway

Oracle Transparent Gateway for IBM DRDA enables Oracle applications to access the DRDA Application Servers, such as DB2 for MVS, through Structured Query Language (SQL). The gateway and Oracle9i server together create the appearance that all data resides on a local Oracle9i 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.
Gateway Capabilities

Oracle Transparent Gateway for IBM 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 IBM DRDA sets a high standard for seamless, enterprise-wide information access.

Oracle Transparent Gateway for IBM 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 the data’s physical location or storage characteristics. This transparency not only allows you to integrate heterogeneous data seamlessly, it simplifies your gateway implementation, application development, and maintenance.

Transparency at All Levels

The Oracle Transparent Gateway for IBM 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.
Extended Database Services

Following are some of the more sophisticated Oracle9i server services available through the gateway.

- **SQL functions**
  
  Your application can access all your data using Oracle SQL, which is rich in features. Advanced Oracle9i 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 Oracle9i 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 Oracle9i 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 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 your data, ensuring uniform enforcement of your business rules across the enterprise.
Extended Advanced Networking, Internet and Intranet Support

The gateway integration with the Oracle9i 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 Oracle9i 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.

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 IBM 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.
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.

Data Definition Language

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

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.

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.
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.

Oracle Stored Procedures

Oracle stored procedures enable you to access and update DB2 data using centralized business rules stored in the Oracle9i 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.

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.

Languages

Any application or tool that supports the Oracle9i 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.

Oracle Server Technology and Tools

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

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

Two-Phase Commit and Multisite Transactions

The gateway can participate as a partner in multisite 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 IBM 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 13-9 in Chapter 13, "Using the Gateway".

Site Autonomy

All Oracle 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.

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 server technology is non-intrusive, providing coexistence and an easy migration path.
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 server, as described in Oracle9i Database Administrator’s Guide. Oracle database security is mapped to the data dictionary of the data source.

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 Oracle9i server instance that communicates with the Oracle Transparent Gateway for IBM 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 implementations of DB2. DB2/UDB is frequently used as an abbreviation for DB2 Universal Database.
The Oracle Transparent Gateway for IBM DRDA works with the Oracle9i server to shield most of the differences of the non-Oracle database from Oracle applications. This means that the Oracle applications can access the Oracle9i 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 Oracle9i 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 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 Oracle9i, you must install Oracle Net on the Oracle9i server machine.

- **Oracle Transparent Gateway for IBM DRDA**
  The gateway must be installed on hosts 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 Oracle9i server can communicate.

- **DRDA server**
  The DRDA server must be a DB2/OS390, DB2/400, DB2 Universal Database, or DB2 server for VM database on a system accessible to the host using either the SNA or TCP/IP protocols.

Multiple Oracle9i 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 described above.
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 IBM DRDA

The gateway does not have:

- Control, redo log, or database files
- The full set of subdirectories and ancillary files associated with an installed Oracle9i 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 Oracle9i 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.

How the Gateway Works

The gateway has no database functions of its own. Instead, it provides an interface by which an Oracle9i server can direct part or all of a SQL operation to a DRDA database.

The gateway supporting the DRDA server is identified to the Oracle integrating server using a database link. The database link is the same construct used to identify other Oracle9i server databases. Tables on the DRDA server are referenced in SQL as:

\[ \text{table}_\text{name}@\text{dblink}_\text{name} \]

or

\[ \text{owner}\.\text{table}_\text{name}@\text{dblink}_\text{name} \]

If you create synonyms or views in the Oracle integrating server database, then you can refer to tables on the DRDA server 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 on the DRDA server, the applicable portion of the SQL statement is sent to the gateway for processing. Any host variables 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. The Oracle integrating server and the application read and process only Oracle datatypes.
**SQL Differences**

Not all SQL implementations are the same. The Oracle9i server supports a larger set of built-in functions than the databases 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 Oracle9i server function can be converted to a function recognizable to the specific DRDA server. For example, the Oracle9i 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.

**Oracle Tools and the Gateway**

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

While the Oracle Transparent Gateway for IBM 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.

The gateway used 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.

**SQL*Plus**

Use SQL*Plus and the Oracle Transparent Gateway for IBM 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.
Features

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

Heterogeneous Services Architecture

This release of the Oracle Transparent Gateway for IBM DRDA utilizes the Oracle Heterogeneous Services component within the Oracle9i server. Heterogeneous Services is the building block for the next generation of Oracle Open Gateways.

For detailed information about heterogeneous services, refer to Oracle9i Heterogeneous Connectivity Administrator’s Guide.

Performance Enhancements

Oracle Transparent Gateway for IBM 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 8 gateways. The actual performance improvement at your site might vary, depending on your installation type and workload.

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 Oracle9i Heterogeneous Connectivity Administrator’s Guide for more information.

Oracle 9i Passthrough Supported

You can use the Oracle9i DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE feature to pass commands or statements available in your DRDA database through the gateway.
Retrieving Result Sets Through Passthrough

Oracle Transparent Gateway for IBM DRDA provides a facility to retrieve result sets from a select SQL statement issued with passthrough. Refer to "Retrieving Result Sets Through Passthrough" on page 14-38 for additional information.

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.

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 14-24 for further information.

Columns Supported in a Result Set

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

EXPLAIN_PLAN Improvement

The EXPLAIN_PLAN table contains the actual SQL statements passed to the DRDA server from the Oracle9i server through the gateway.

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.
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.

Large Base of Data Access

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

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.
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.
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.

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

- Changes and Enhancements, Release 9.2.0.1.0 on page 2-2
- Known Problems on page 2-5
- Known Restrictions on page 2-6
Product Set

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

<table>
<thead>
<tr>
<th>Product</th>
<th>Release number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Transparent Gateway for IBM DRDA</td>
<td>9.2.0.1.0</td>
</tr>
<tr>
<td>Oracle Net</td>
<td>9.2.0.1.0</td>
</tr>
</tbody>
</table>

Changes and Enhancements, Release 9.2.0.1.0

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

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.

Enhancements in Release 9.0.1

Enhancements

Support for DB2/OS390 V6 and V7 Stored Procedures

This release of the gateway supports the native stored procedure catalogs in DB2 V6 and DB2 V7 (SYSIBM.SYSROUTINES and SYSIBM.SYPARMS).

Codepage Map Facility

This release of the gateway supports external mapping of IBM CCSIDs to Oracle character sets. Refer to "Gateway Codepage Map Facility" in Appendix D, "National Language Support".

Improved Oracle Data Dictionaries

This release of the gateway ships with a set of improved data dictionary tables and views which help the gateway better integrate with the Oracle9i server.
Changes in Release 9.0.1

Following is a list of changes unique to release 9.0.1 of the gateway.

**Oracle Server Dependencies**

This release of the Oracle Transparent Gateway for IBM DRDA requires the latest released patch set for Oracle9i server release 9.2 or release 9.0.1, or Oracle8i server release 8.1.7.

**Gateway Parameters**

The syntax of `initsid.ora` has been simplified. Refer to Appendix C for details.

Refer to the "Heterogeneous Services" section of Oracle9i Heterogeneous Connectivity Administrator’s Guide for complete information about heterogeneous services.

**TNSNAMES.ORA**

The service name definition (from the Oracle integrating server to the gateway) must contain an extra parameter, (HS=). Refer to Chapter 11, "Configuring the Gateway", for additional information.

**IBM DB2 Universal Database Support**

This release supports IBM DB2 Universal Database.

**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 IBM DRDA supports access to EBCDIC tables and ASCII tables. Refer to Appendix D, "National Language Support".

**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-12.
<table>
<thead>
<tr>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1947548</td>
<td>QA - IN DOUBT TRANSACTION FOR SNA/DRDA WHEN DROPPING AND CREATING TABLES</td>
</tr>
<tr>
<td>1941672</td>
<td>QA - ORA-1821 CAN NOT INSERT USING TO_DATE () THROUGH GATEWAY</td>
</tr>
<tr>
<td>1927573</td>
<td>QA - COMMENTS IN O2PC.SQL ARE INCORRECT</td>
</tr>
<tr>
<td>1901348</td>
<td>QA - SELECT USING BIND VARIABLE IN TO_DATE FUNCTION FAILS WITH ORA-28511</td>
</tr>
<tr>
<td>1891118</td>
<td>ADD SUPPORT FOR DATA CONNECTION' SNAP-IX SNA PRODUCT</td>
</tr>
<tr>
<td>1878053</td>
<td>QA - ORA-28522 ERROR INITIALIZING HETEROGENEOUS CAPABILITIES</td>
</tr>
<tr>
<td>1806810</td>
<td>ORA-28511 on SELECT TO_DATE with bind variable</td>
</tr>
<tr>
<td>1755771</td>
<td>ORA-2057 in alert log when gateway connection fails</td>
</tr>
<tr>
<td>1755080</td>
<td>ORA-00942 and ORA-28506 querying data dictionary views</td>
</tr>
<tr>
<td>1729291</td>
<td>Support for character set EL8ISO8859P7 required for TG4DRDA for Solaris</td>
</tr>
<tr>
<td>1665037</td>
<td>ORA-04053 When DRDA_DISABLE_CALL=FALSE using oracle stored procedure</td>
</tr>
<tr>
<td>1422130</td>
<td>Support for character set AR8ISO8859P6 required for the Transparent Gateway for IBM DRDA on NT.</td>
</tr>
<tr>
<td>1376928</td>
<td>TG4DRDA does not support drda servers using code page 1252 (WE8MSWIN1252)</td>
</tr>
<tr>
<td>1366454</td>
<td>There was a Kernel core dump when running consecutive stored procedures to DB2/OS390.</td>
</tr>
<tr>
<td>1358377</td>
<td>ORA-28500 and DB2 error -301 received when using gateway with microfocus cobol.</td>
</tr>
<tr>
<td>1353699</td>
<td>Gateway debug version returned errmc=124c.</td>
</tr>
<tr>
<td>1336994</td>
<td>DB2 threads remained and TCP connection was still established after exiting Plus.</td>
</tr>
<tr>
<td>1255848</td>
<td>FDS_CLASS in DRDAAS400.sh</td>
</tr>
<tr>
<td>1255698</td>
<td>Binding FDRDA gateway package</td>
</tr>
<tr>
<td>1211368</td>
<td>-30020 Trying to connect to AS/400 using TCP/IP with DRDA</td>
</tr>
</tbody>
</table>
Table 2–1  Bugs Fixed in Release 9 of the Gateway

<table>
<thead>
<tr>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1149824</td>
<td>Deinstalling Oracle Transparent Gateway for IBM DRDA.</td>
</tr>
<tr>
<td>1033074</td>
<td>Port of base bug 661051 to Oracle Transparent Gateway for IBM DRDA product.</td>
</tr>
<tr>
<td>992511</td>
<td>Any statement hangs on SQL*Plus via tg4drda</td>
</tr>
<tr>
<td>940538</td>
<td>Oracle Transparent Gateway for IBM DRDA would hang when executing GTW$_BIND_PKG bind package.</td>
</tr>
<tr>
<td>917951</td>
<td>Unsuccessful execution deadlock or timeout</td>
</tr>
<tr>
<td>903732</td>
<td>Update to table in fds with large number</td>
</tr>
<tr>
<td>887219</td>
<td>Insert NULL to DB2</td>
</tr>
</tbody>
</table>

Known Problems

The problems documented in the following section are specific to the Oracle Transparent Gateway for IBM 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 for Microsoft Windows or DB2/UDB for Linux.
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.

**SQL*Plus DESCRIBE command**

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

**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 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.

**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 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.
Known Restrictions

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.

Character Set Limitation
Oracle Transparent Gateway for IBM DRDA does not support access to double-byte character set (DBCS) data.

Datatype Limitations
Refer to ”DRDA Datatype to Oracle Datatype Conversion” on page 14-26 for detailed information about datatypes.

SAVEPOINT Command Is Not Supported
Oracle Transparent Gateway for IBM 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 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 Oracle server’s DB_DOMAIN parameter. Refer to Chapter 11, ”Configuring the Gateway”, for more information.
Known Restrictions

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/UID restriction, the ORACLE2PC table must be created in the DB2/UID database before the package can be bound. For details, refer to Chapter 11, "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:

\[ \text{date + number} \]
\[ \text{number + date} \]
\[ \text{date - number} \]
\[ \text{date1 - date2} \]

DRDA servers do not allow number addition or subtraction with date datatypes. The date and number addition and subtraction \((\text{date + number}, \text{number + date}, \text{date - number})\) 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 \((\text{date1 - date2})\), 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 IBM DRDA.
Dictionary Views Are Not Provided for DB2/VM or DB2/UDB

Currently the Oracle Transparent Gateway for IBM DRDA provides SQL for defining DB2/OS390 and DB2/400 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 and DB2/UDB 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’s Initialization Parameters must occur at this point 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.

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 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 IBM DRDA does not support CONNECT BY in SELECT statements.
This chapter provides information about hardware and software requirements that is specific to this release of the Oracle Transparent Gateway for IBM DRDA. It includes the following sections:

- Hardware Requirements on page 3-2
- Software Requirements on page 3-3
- Documentation Requirements on page 3-5
Hardware Requirements

Includes processor, memory, network attachment, and drives

Processor

This gateway requires a host with or without multiple processors that is capable of running 64-bit applications.

The UNIX platforms supported by this gateway release are: Sun Solaris Operating System (SPARC), IBM RS/6000 AIX-based system, Hewlett-Packard 9000 Series 700 or 800.

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 IBM 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

Network Attachment

The hardware requires any network attachment 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. The hardware must support independent LUs if you want concurrent SNA access. The network attachment for TCP/IP is typically an Ethernet attachment.

CD-ROM Drive

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

Disk space required for installation:
Solaris requires 550 MB
AIX requires 1.1 GB
HP-UX requires 550 MB

Software Requirements

The system software configuration 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.

Operating System

- Solaris 2.6 or later
- AIX 4.3.3 or AIX 5L
- HP-UX 11.0 (64-bit)

DRDA Databases

You must have at least one of the following IBM DRDA databases at a release level supported by IBM:
- DB2/OS390
- DB2/VM
- DB2/400
- DB2/Universal Database
Communication Server

The operating systems utilize specific communications servers.

**Solaris**

Sunlink SNA Peer-to-Peer Communications Server, Version 9
SNAP-IX, Version 6

**AIX**

IBM Communications Server for AIX, Version 6.0

**HP**

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

Oracle Server

The Oracle server which is to act as the Oracle integrating server requires the latest released patch set for Oracle9i server release 9.2 or release 9.0.1, or Oracle8i server release 8.1.7.

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 Oracle9i Server:

- Oracle Net Client 9.2.0.1.0
- an Oracle Adapter version 9.2.0.1.0

Oracle Net software is included in this Oracle Transparent Gateway for IBM 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.
Documentation Requirements

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

- Oracle9i Database Administrator’s Guide
- Oracle9i Application Developer’s Guide - Fundamentals
- Oracle9i Heterogeneous Connectivity Administrator’s Guide
- Oracle9i Database Error Messages
- Oracle C++ Call Interface Programmer’s Guide
- Oracle Call Interface Programmer’s Guide
- SQL*Plus User’s Guide and Reference
- Oracle9i SQL Reference
- Oracle9i Net Services Administrator’s Guide
- Oracle9i 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.
This chapter provides general information about gateway installation that is specific to this release of the Oracle Transparent Gateway for IBM 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 4-8
Introduction

The complete Oracle Transparent Gateway for IBM DRDA installation process is divided into installation and configuration tasks. This process is described in Chapters 4 through 8. If this is the first time the gateway has been installed on your host, then you must perform all of the steps 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.

Before You Begin

This chapter requires you to input parameters unique to your system in order to properly configure the gateway. Refer to Appendix E, "Configuration Worksheet" for a worksheet listing all the installation parameters you will need to know 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.
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
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 11, "Configuring the Gateway".

For general information about installing Oracle products, and how to use the Oracle Universal Installer, refer to the Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris.

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.

Installing the Gateway from CD-ROM

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

Step 1: Log on to the host

Log on to your host as the Oracle database administrator (DBA) user. Refer to the preinstallation chapter in the Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris.
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:

```bash
$ mkdir /oracle
$ mkdir /oracle/tg4drda
$ mkdir /oracle/tg4drda/9.2.0
$ chown oracle:dba /oracle/tg4drda/9.2.0
$ chmod 755 /oracle/tg4drda/9.2.0
```

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:

```bash
$ ORACLE_HOME=/oracle/tg4drda/9.2.0; export ORACLE_HOME
```

If you are a C shell user, then enter:

```bash
$ setenv ORACLE_HOME /oracle/tg4drda/9.2.0
```

Step 4: Mount the CD-ROM

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

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:

```bash
$ su root
# mkdir /cdrom
# mount -r -F hsfs /dev/dsk/c0t6d0s0 /cdrom
# exit
$ cd /cdrom
```
Installing the Gateway from CD-ROM

Step 4b: on AIX, enter:

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

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

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

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 Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris.

Start the Installer with the following command:

$ ./runInstaller
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 table as a guide to the installation. The left column lists the prompts that Oracle Universal Installer offers; respond to the prompts by implementing the actions listed in the corresponding "Response" column on the right.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome</td>
<td>Click &quot;Next&quot;.</td>
</tr>
<tr>
<td>File Locations: &quot;Source and Destination&quot;</td>
<td>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 &quot;Next&quot;.</td>
</tr>
<tr>
<td>Available Products</td>
<td>Select &quot;Oracle 9i Database&quot;. Click &quot;Next&quot;.</td>
</tr>
<tr>
<td>Installation Types</td>
<td>Select &quot;Custom&quot;. Click &quot;Next&quot;.</td>
</tr>
<tr>
<td>Available Products Components</td>
<td>Open the &quot;Oracle Transparent Gateways&quot; product group and select &quot;Oracle Transparent Gateway for IBM DRDA&quot;. Remove selection from everything else for a standalone gateway installation. Click &quot;Next&quot;.</td>
</tr>
<tr>
<td>Optional JDK home prompt</td>
<td>Click &quot;Next&quot;.</td>
</tr>
<tr>
<td>DRDA Network Interface Product Software</td>
<td>Choose the network interface software appropriate for this installation of the gateway. Click &quot;Next&quot;.</td>
</tr>
<tr>
<td>Summary</td>
<td>Verify the products to be installed. Click &quot;Next&quot;.</td>
</tr>
</tbody>
</table>

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.
Installation Complete

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

De-installing the Gateway

De-installing the Oracle Transparent Gateway for IBM 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 5-5
- DB2/UDB (Universal Database) on page 5-7
- DB2/VM on page 5-9

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.
Checklists for Configuring the DRDA Server

Use the following checklists for configuring the DRDA server.

**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

**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

**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

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

**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.

**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 both DB2 and OS/390 (MVS).
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.

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.

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

**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.

**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.

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

**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.

**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 rebinding 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 both the DB2 instance ID and AIX.
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.

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

**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.

**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.

**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.
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.
This chapter describes configuring the SunLink SNA Peer-to-Peer product on Solaris for use with the Oracle Transparent Gateway for IBM 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.

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 6-6
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
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.

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

**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.

**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.
**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.

**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.

**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.
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'

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.

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.
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.

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.
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 4
- Configuring SNAP-IX on page 4
- Testing the Connection on page 19
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

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.

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.

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.

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
Creating SNA Definitions for the Gateway

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.

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.
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.

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.

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)
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].
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.
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 affect initial XID contact and retransmission times and limits. The defaults are normally sufficient. Click [OK].
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].
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].
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.
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].
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.
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
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.

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.

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.

**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.

**Testing the Connection**

Before proceeding with the gateway configuration tasks in Chapter 11, "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
This chapter describes configuring the IBM Communication Server product on AIX for usage with the Oracle Transparent Gateway for IBM 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 8-3
Step 2: Creating Communication Server Profiles for the Gateway on page 8-3
Step 3: Testing the Connection on page 8-6
Using SNA Session Security Validation on page 8-7
SNA Conversation Security on page 8-8
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
Step 1: Configuring Communication Server Profiles

Configure the profiles to define APPC conversations with DRDA databases.

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.

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.

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.

**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.

**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.
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.

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.

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.

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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local LU Name</td>
<td>LU Name as specified in local LU profile</td>
</tr>
<tr>
<td>Fully Qualified Partner LU Name or partner LU alias</td>
<td>LU Name specified in Partner LU profile or Partner LU location profile</td>
</tr>
<tr>
<td>Mode Name</td>
<td>Mode Name specified in Mode profile</td>
</tr>
<tr>
<td>Remote Transaction Program Name</td>
<td>Remote TPN</td>
</tr>
<tr>
<td>Remote TPN in hexadecimal</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.

**Step 3: Testing the Connection**

Before proceeding with the gateway configuration tasks in Chapter 8, "Configuring the Gateway", ensure that your connection is working. This can be done using SMIT.
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.
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.

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.
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 -1` 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.
This chapter describes configuring the SNAPlus2 product on HP-UX for usage with the Oracle Transparent Gateway for IBM DRDA. SNAPlus2 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
- SNAPlus2 Configuration Tool on page 9-2
- Creating SNAPlus2 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 9-18
- SNA Conversation Security on page 9-19
- Testing the Connection on page 9-21
Steps for Configuring the Communications Interfaces

1. Create SNAPlus2 profiles for the gateway
2. Create SNA definitions for the gateway
3. Test the connection

Before You Begin

This chapter requires you to input parameters unique to your system in order to properly configure SNAPlus2. 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.

SNAPlus2 Configuration Tool

All SNAPlus2 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 SNAPlus2 configuration and the current running state of the host SNA node. Refer to the HP-UX SNAPlus2 administrators guide for more details on using xsnapadmin.

Creating SNAPlus2 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.

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 SNMPPlus2. 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.

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.
Sample SNAPlus2 Definitions

The $ORACLE_HOME/tg4drda/sna/snaplus subdirectory contains a set of sample SNAPlus2 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 provided will not work without being tailored for the local host and SNA network.

Configuring SNAPlus2

This section describes the process of creating your SNA definitions for SNAPlus2, 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.

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)
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].
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].
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. 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 \texttt{lanscan} command. Click \texttt{[OK]}. 

\textbf{Figure 9–3  Add to <nodename> 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].
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].

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.
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].
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].
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].
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.
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].
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.
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
Using SNA Session Security Validation

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 SNAPlus2 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.
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.

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.
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, SNAPlus2 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.
Testing the Connection

Before proceeding with the gateway configuration tasks in Chapter 11, "Configuring the Gateway", ensure that your connection is working. Do this by starting the SNAPlus2 Node and then starting the individual link stations.

Figure 9–15 shows the relationship between SNAPlus2 definitions and the VTAM definitions on the remote host.
Figure 9–15  Relationship between SNAPlus2 Definitions and Host VTAM Definitions
This chapter describes configuring TCP/IP for the various Unix platforms supported by the Oracle Transparent Gateway for IBM 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
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.

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.
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 11-2
- Choosing a Gateway System Identifier (SID) on page 11-4
- Gateway Configuration and the Startup Shell Script on page 11-4
- Configuring the Host on page 11-4
- DRDA Gateway Package Considerations on page 11-10
- Backup and Recovery of Gateway Configuration on page 11-12
- Configuring the Oracle Integrating Server on page 11-13
- Accessing the Gateway from Other Oracle Servers on page 11-13
- Accessing Other DRDA Servers on page 11-14
- Gateway Installation and Configuration Complete on page 11-14
Configuration Checklists

Configuring the Gateway

- Choosing a Gateway System Identifier (SID)

Configuring the Host

- Step 1: Choose an appropriate Startup Shell Script
- Step 2: Create the gateway Startup Shell Script
- Step 3: Tailor the Startup Shell Script
- Step 4: Choose the initsid.ora file
- Step 5: 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 or DB2/400, 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 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.
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 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 11-14 for information on configuring additional SIDs.

Gateway Configuration and the Startup Shell Script

The data in this chapter describe the configuration process for the gateway. You should notice that most, if not all, gateway parameters may be moved into the initsid.ora initialization file, which would allow you to omit the Startup Shell Script. However, the Startup Shell Script provides an environment for troubleshooting and debugging if any problems occur with the gateway. The only advantage of removing the Startup Shell Script is a small decrease in gateway startup time. If this decrease in startup time is important to you in your usage of the gateway, then please contact Oracle Support Services for details on further configuration in this manner.

Configuring the Host

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

Step 1: Choose an appropriate Startup Shell Script

In previous versions of the gateway, the initialization parameters were stored in files named initsid.ora and initsid.gtwboot in the gateway instance directories. With version 9i of the gateway, most parameters that were in initsid.gtwboot have been moved to a Startup Shell Script file. The syntax of the Startup Shell Script is that of the UNIX Bourne shell. The syntax of the initsid.ora has been simplified. Refer to Appendix C for details.
When migrating from previous releases of Oracle Transparent Gateway for IBM DRDA, please be aware of these differences.

First, choose a model startup shell script. Samples are stored in the $ORACLE_HOME/tg4drda/admin directory. The following list shows the startup shell scripts for various DRDA server platforms:

- sample startup shell script for DB2/OS390: drdaDB2.sh
- sample startup shell script for DB2/UDB: drdaDB2UDB.sh
- sample startup shell script for DB2/400: drdaAS400.sh
- sample startup shell script for DB2/VM: drdaDB2VM.sh

Step 2: Create the gateway Startup Shell Script

In previous versions of the gateway, a startup program was provided, named g4drdrv. This program was specified at the PROGRAM parameter in the listener.ora file. This program also was responsible for reading the initsid.gtwboot and starting the gateway server itself.

In version 9i of the gateway, this process has been moved to a shell script. Most of the init parameters that were stored in the initsid.gtwboot are now stored in this Startup Shell Script, and the name of the script will now be used as the PROGRAM parameter in the listener.ora file. Sample scripts, named ‘drdaXXX.sh’ are stored in the $ORACLE_HOME/tg4drda/admin directory. Copy one to a name of your choosing and move it to the $ORACLE_HOME/bin directory.

After an appropriate model of startup shell script has been chosen, copy it to a name of your choosing in the $ORACLE_HOME/bin directory. For example:

```
$ cp $ORACLE_HOME/tg4drda/admin/drdaDB2.sh $ORACLE_HOME/bin/drdahoa1.sh
```

**Hint:** Oracle suggests naming the shell script the same as the SID for the gateway in order to make name associations easier.
Configuring the Host

Step 3: Tailor the Startup Shell Script

The Startup Shell Script file contains parameters that govern:

- The UNIX environment in which the gateway process runs, including NLS-related parameters. See Appendix D for information on tailoring the parameters for NLS.
- The environment variable settings needed by SNA server software to initialize the gateway APPC interface.

Customize the file you created in Step 2 as needed to tailor the Startup Shell Script file for your installation.

NOTE: If you are using TCP/IP, then do not perform the instructions for changing environmental variable settings for SNA.

Step 4: 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.
Step 5: 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 & 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 for a description of each parameter:

- DRDA_CONNECT_PARM
- DRDA_PACKAGE_COLLID
- DRDA_PACKAGE_NAME
- DRDA_PACKAGE_OWNER
- DRDA_REMOTE_DB_NAME
- HS_DB_NAME
- HS_DB_DOMAIN
Configuring the Host

## Binding the DRDA Gateway Package

The product requires a package to be bound on the DRDA server. The gateway has an internal, stored procedure which 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 12-4. Also refer to "Configuring the Oracle Integrating Server" in this chapter on page 11-13.) Before this package can be bound on the DRDA server, the Gateway Initialization File must be correctly configured, and a Startup Shell Script must be chosen and configured (refer to Appendix C, "DRDA-Specific Parameters").

1. Log on to an Oracle integrating server.
   
   Use either SQL*Plus or Server Manager:
   
   ```bash
   $ 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
   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 11-13 for more information.

3. Execute the stored procedure GTW$_BIND_PKG:
   
   ```sql
   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.
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 that 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.

4. Create the ORACLE2PC table:
   $ db2 -tf o2pc.sql

5. Commit the transaction:
   $ db2 'COMMlT'

6. Verify that the table was created.
   Optionally, verify the table was created under the correct user ID:
   
   $ db2 'LIST TABLES FOR USER'
   $ db2 'COMMlT'
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 4 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".

Before Binding the DRDA Gateway Package

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

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.

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 11-9 for information on creating the table.

7. Disconnect from the session:

$ db2 'DISCONNECT CURRENT'
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.

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

If your DRDA server is DB2/OS390 or DB2/400, 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 `g4dtab.sql` and `g4dview.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.

Backup and Recovery of Gateway Configuration

The configuration of the gateway is stored in the Gateway Initialization File and in the Startup Shell Script. These are stored in $ORACLE_HOME/tg4drda/admin and $ORACLE_HOME/bin, respectively. Since they are simple files, you may back them up using an archiving tool of your choice.
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.

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 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 12, "Oracle Net", "Step 2: Modify tnsnames.ora file" on page 12-5).

```
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.

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 13-5 for information.

Accessing the Gateway from Other Oracle Servers

Perform the following steps for each of the Oracle 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 servers to the gateway.
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 Startup Shell Scripts and 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.

Gateway Installation and Configuration Complete

The Oracle Transparent Gateway for IBM DRDA installation and configuration process is now complete. The gateway is ready for use.
Oracle Net is an Oracle product providing network communication between Oracle applications, Oracle Servers and Oracle Gateways across different systems.

This chapter contains the following sections:

- Checklists for Oracle Net on page 12-2
- Oracle Net Introduction on page 12-3
- Oracle Net Overview on page 12-3
- Configuring Oracle Net on page 12-4
- Advanced Security Encryption on page 12-6
- Setting Up Advanced Security Encryption for Test on page 12-7
- Testing Advanced Security Encryptions on page 12-8
Checklists for Oracle Net

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

Configuring Oracle Net

- Step 1: Modify listener.ora file
- Step 2: Modify tnsnames.ora file

Advanced Security Encryption

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

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 Integrating Server

Testing Advanced Security Encryptions

- Step 1: Connect Gateway and Oracle Integrating Server
- Step 2: Reset Configuration Parameters on the Gateway
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 Oracle9i Net Services Administrator’s Guide, Oracle9i Net Services Reference Guide, and Oracle9i Heterogeneous Connectivity Administrator’s Guide.

Oracle Net Overview

Oracle Net is a required Oracle product supporting network communications between Oracle applications, Oracle 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 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.

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 server or Oracle gateway.

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 server running on a host computer can share and obtain information retrieved from other remote Oracle servers. Regardless of the number of database information sources, you might be aware of only one logical database.
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 server or gateway.
- **client (task)** is the application using a Oracle Net driver to communicate with the Oracle 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.

Configuring Oracle Net

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

**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/9.2.0)
  (PROGRAM=drdahoa1.sh))
```

Refer to Appendix B, "Sample Files", for a sample listener.ora file.

**Note:** The PROGRAM=drdahoa1.sh parameter is required. It specifies to the listener the name of the gateway Startup Shell Script.
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 Oracle9i 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 Oracle9i Net Services Administrator’s Guide.

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.
Setting Up Advanced Security Encryption for Test

The following procedures test Advance 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

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 = "abcdefg123456789"
```

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.

Step 2: Set Advanced Security Encryption Parameters for 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 = "abcdefg123456789"
```

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

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

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.
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 Oracle9i Administrator’s 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 13-4
- Accessing AS/400 File Members on page 13-5
- Using the Synonym Feature on page 13-5
- Performing Distributed Queries on page 13-6
- Read-Only Gateway on page 13-9
- Replicating in a Heterogeneous Environment on page 13-9
- Copying Data from the Oracle9i Server to the DRDA Server on page 13-10
- Copying Data from the DRDA Server to the Oracle9i Server on page 13-10
- Tracing SQL Statements on page 13-11
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.

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 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.

`userid` 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.
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.

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 Oracle9i Administrator’s Guide for additional information about dropping database links.

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.

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 Oracle9i Administrator’s Reference for additional information about limiting the number of active database links.
Accessing the Gateway

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

Step 1: Login to the Oracle Integrating Server
Login

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’
```

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 V9.2.0.1.0 grc=0, drc=-777 (83TC,0000), errp=ARIX0,
sqrcode=-551, sqlstate=42501, errd=FFFFFF9C,0,0,0,0,0
errmc=USER SELECT SECURE.TABLE
Accessing AS/400 File Members

There is nothing specific to DRDA or to the gateway that 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.

Using the Synonym Feature

You can provide complete data, location, and network transparency by using the synonym feature of the Oracle 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 Oracle9i Administrator’s 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
Performing Distributed Queries

The Oracle Transparent Gateway technology enables the execution of distributed queries that join Oracle 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-8 for more information.
Example of a Distributed Query

The following example joins data between an Oracle 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:

<table>
<thead>
<tr>
<th>CUSTNAME</th>
<th>PROJNO</th>
<th>ENAME</th>
<th>SPEND</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC Co.</td>
<td>1</td>
<td>Jones</td>
<td>400</td>
</tr>
<tr>
<td>ABC Co.</td>
<td>1</td>
<td>Smith</td>
<td>180</td>
</tr>
<tr>
<td>XYZ Inc.</td>
<td>2</td>
<td>Jones</td>
<td>400</td>
</tr>
<tr>
<td>XYZ Inc.</td>
<td>2</td>
<td>Smith</td>
<td>180</td>
</tr>
</tbody>
</table>
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 server supports the PREPARE TRANSACTION statement. Any number of Oracle 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 11-10 for more information.

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 Oracle9i Administrator's Guide.
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.

Replicating in a Heterogeneous Environment

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

Oracle9i Server Triggers

When updates are made to the Oracle server, synchronous copies of Oracle and non-Oracle data can be maintained automatically by using Oracle9i server triggers.

Oracle Snapshots

Oracle Transparent Gateway for IBM DRDA can use the Oracle snapshot feature to automatically replicate non-Oracle data into the Oracle 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 server at user-defined intervals.
Copying Data from the Oracle9i Server to the DRDA Server

The COPY command enables you to copy data from an Oracle 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
```

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@ORACLE9 -
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.

Copying Data from the DRDA Server to the Oracle9i Server

The CREATE TABLE command enables you to copy data from a DRDA server database to an Oracle 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 server. For more information about the COPY command, refer to the SQL*Plus User’s Guide and Reference.
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 on 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 17-8 in Chapter 17, "Error Messages, Diagnosis, and Reporting".
Developing Applications

The Oracle Transparent Gateway for IBM DRDA allows applications written for the Oracle 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, there are fundamental SQL, datatype, and semantic differences between the Oracle 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 IBM 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 14-5
- Database Link Behavior on page 14-7
- Oracle Server SQL Construct Processing on page 14-8
- Native Semantics on page 14-24
- DRDA Datatype to Oracle Datatype Conversion on page 14-26
- Passing Native SQL Statements through the Gateway on page 14-36
- Oracle Data Dictionary Emulation on a DRDA Server on page 14-38
- Defining the Number of DRDA Cursors on page 14-39
Gateway Appearance to Application Programs

An application 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 defined at the Oracle integrating server. Your application specifies tables that exist on a DRDA database using the name 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 in your application:

  ```sql
  SELECT EMPNO, SALARY
  FROM EMP, EMPS@DRDA
  WHERE
  ```

  In this example, `EMP` is a table on an Oracle 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.
Fetch Reblocking

The Oracle 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 Oracle9i server. The buffer might contain one or more qualified rows from the DRDA server. This feature can provide significant performance enhancements, depending on your application design, installation type, and workload.

The array size between the client and the Oracle9i server is still determined by the Oracle application.

Refer to Chapter 11, "Configuring the Gateway", for more information.

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 server and 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.
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.
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 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 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 than invoking a stored procedure at a remote Oracle instance.
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 to calculate a new yearly salary for JOHN SMYTHE. The revised salary returned in RESULT is used to update the EMP table of an Oracle database server:

```sql
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’s 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.SYSPARAMS, and DB2/400 uses the table QSYS2.SYSPROCS and QSYS2.SYSPARAMS. The gateway has a list of known catalogs to search, depending upon the DRDA server 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 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.

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 like a VSAM file, then the gateway considers 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 passed to and from the DB2 stored procedure cannot be null.

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.

Oracle 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

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 server and DRDA database. These SQL constructs are forwarded unmodified. All the compatible functions are column functions. Functions that are not compatible are either translated to an equivalent DRDA SQL function or are compensated by the Oracle server after the data is returned from the DRDA database.

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 server and then forwarded to the DRDA database. The Oracle integrating server, transparent to your application, changes the function name before sending it to the DRDA database.
Compensated SQL Functions

Some advanced SQL constructs that are supported by the Oracle 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.

The Oracle server can compensate for the missing or incompatible function by automatically excluding the incompatible SQL construct from the SQL request that is forwarded to the DRDA database. The Oracle 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 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 stored in a DRDA database, then keep in mind that some functions require post-processing.

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). Refer to “Native Semantics” on page 14-24 for more information.
DB2/OS390 SQL Compatibility

The ways that the Oracle server and gateway handle SQL functions for a DB2/OS390 database are shown in the following table:

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ACOS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD_MONTHS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ASIN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAN2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEIL</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CHARTOROWID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CONCAT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COS</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COSH</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COUNT(*)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECODE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FLOOR</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GREATEST</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Table 14–1  DB2/OS390 SQL Compatibility, by Oracle SQL Function

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEXTORAW</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INITCAP</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INSTR</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INSTRB</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LAST_DAY</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LEAST</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTHB</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LOG</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LOWER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LPAD</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LTRIM</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MOD</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MONTHS_BETWEEN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NEW_TIME</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NEXT_DAY</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NLS_INITCAP</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NLS_LOWER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NLS_UPPER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NLSSORT</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NVL</td>
<td></td>
<td></td>
<td>VALUE</td>
<td></td>
</tr>
</tbody>
</table>
Table 14–1  DB2/OS390 SQL Compatibility, by Oracle SQL Function

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RAWTOHEX</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUND</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROWIDTOCHAR</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPAD</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RTRIM</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIN</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SINH</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SOUNDEX</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQRT</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>STDDEV</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTRB</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUM</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDATE</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAN</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TANH</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_CHAR</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_DATE</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_LABEL</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_MULTI_BYTE</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_NUMBER</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The ways that the Oracle 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

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO_SINGLE_BYTE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSLATE</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TRUNC</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>UPPER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>USER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERENV</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VARIANCE</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VSIZE</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### DB2/Universal Database SQL Compatibility

The ways that the Oracle 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

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACOS</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD_MONTHS</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ASIN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ATAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAN2</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 14–2  DB2/Universal Database SQL Compatibility, by Oracle SQL Function

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEIL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHARTOROWID</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONCAT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COSH</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COUNT(*)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECODE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOOR</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GREATEST</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HEXTORAW</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INITCAP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTR</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INSTRB</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LAST_DAY</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEAST</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTHB</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOG</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LOWER</td>
<td></td>
<td>LCASE</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Oracle SQL Function</td>
<td>Compatible</td>
<td>Translated</td>
<td>Compensated</td>
<td>Native Semantics Candidate</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>LPAD</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LTRIM</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOD</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MONTHS_BETWEEN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NEW_TIME</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NEXT_DAY</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NLS_INITCAP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_LOWER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_UPPER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLSSORT</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVL</td>
<td>VALUE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POWER</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RAWTOHEX</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUND</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROWIDTOCHAR</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RPAD</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTRIM</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SINH</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOUNDSEX</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Oracle SQL Function</td>
<td>Compatible</td>
<td>Translated</td>
<td>Compensated</td>
<td>Native Semantics Candidate</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQRT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STDDEV</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTRB</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUM</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDATE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TANH</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_CHAR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_DATE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_LABEL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_MULTI_BYTE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_NUMBER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_SINGLE_BYTE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSLATE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TRUNC</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPPER</td>
<td>UCASE</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>USER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERENV</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIANCE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VSIZE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DB2/400 SQL Compatibility

The ways that the Oracle 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

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td></td>
<td>ABSVAL</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ACOS</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ADD_MONTHS</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ASIN</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAN</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAN2</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEIL</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CHARTOROWID</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CONCAT</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COS</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>COSH</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>COUNT(*)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECODE</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>FLOOR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GREATEST</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Oracle SQL Function</td>
<td>Compatible</td>
<td>Translated</td>
<td>Compensated</td>
<td>Native Semantics Candidate</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>HEXTORAW</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INITCAP</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INSTR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>INSTRB</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LAST_DAY</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEAST</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTHB</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LOG</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LOWER</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LPAD</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LTRIM</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOD</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MONTHS_BETWEEN</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEW_TIME</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEXT_DAX</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_INITCAP</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_LOWER</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_UPPER</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLSSORT</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NVL</td>
<td></td>
<td>VALUE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 14-3  DB2/400 SQL Compatibility, by Oracle SQL Function

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RAWTOHEX</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUND</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROWIDTOCHAR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPAD</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RTRIM</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SINH</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOUNDEX</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQRT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STDDEV</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBSTR</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTRB</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUM</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDATE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TANH</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_CHAR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_DATE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_LABEL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_MULTI_BYTE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TO_NUMBER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The ways that the Oracle server and gateway handle SQL functions for a DB2/VM database are shown in the following table:

**Table 14–3 DB2/400 SQL Compatibility, by Oracle SQL Function**

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO_SINGLE_BYTE</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSLATE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRUNC</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>UPPER</td>
<td>TRANSLATE</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>USER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>USERENV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIANCE</td>
<td>VAR</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VSIZE</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**DB2/VM SQL Compatibility**

The ways that the Oracle 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**

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ACOS</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD_MONTHS</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ASIN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ATAN</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAN2</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEIL</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Oracle SQL Function</td>
<td>Compatible</td>
<td>Translated</td>
<td>Compensated</td>
<td>Native Semantics Candidate</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>CHARTOROWID</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CONCAT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COSH</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COUNT(*)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECODE</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOOR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GREATEST</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEXTORAW</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITCAP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTRB</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST_DAY</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEAST</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LENGTHB</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LN</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOG</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOWER</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPAD</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 14–4  DB2/VM SQL Compatibility, by Oracle SQL Function

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTRIM</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOD</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MONTHS_BETWEEN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEW_TIME</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEXT_DAY</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_INITCAP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_LOWER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLS_UPPER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLSSORT</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NVL</td>
<td></td>
<td>VALUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POWER</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RAWTOHEX</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>REPLACE</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUND</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROWIDTOCHAR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPAD</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RTRIM</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIGN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SIN</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SINH</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SOUNDEX</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQRT</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Table 14–4  DB2/VM SQL Compatibility, by Oracle SQL Function

<table>
<thead>
<tr>
<th>Oracle SQL Function</th>
<th>Compatible</th>
<th>Translated</th>
<th>Compensated</th>
<th>Native Semantics Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>STDDEV</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUBSTRB</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SUM</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDATE</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TAN</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TANH</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_CHAR</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_DATE</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_LABEL</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_MULTI_BYTE</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_NUMBER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TO_SINGLE_BYTE</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TRANSLATE</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TRUNC</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>UPPER</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>USER</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>USERENV</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VARIANCE</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VSIZE</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Native Semantics

Because some of the advanced SQL constructs that are supported by the Oracle server may not be supported in the same manner (if at all) by the DRDA database, the Oracle server compensates for the missing or incompatible functionality by post-processing the DRDA database data with Oracle (Refer to the previous section, "Oracle Server SQL Construct Processing" on page 14-8 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 functions.

Some of the DRDA servers also provide support for user-defined functions. The user may choose to implement Oracle functions natively, 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. Oracle 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 to the correct 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 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 Oracle 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_CAPABILITY" parameter on page C-3 for details on enabling or disabling these capabilities. Refer to the Oracle9i SQL Reference for the Oracle format of the following capabilities.
SQL Functions Which Can Be Enabled

The following list contains SQL functions that are disabled (OFF) by default. They can be enabled (turned ON) as an option.

<table>
<thead>
<tr>
<th>ABS</th>
<th>ASCII</th>
<th>CEIL</th>
<th>CHR</th>
<th>CONVERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>COSH</td>
<td>DECODE</td>
<td>DUMP</td>
<td>EXP</td>
</tr>
<tr>
<td>FLOOR</td>
<td>GREATEST</td>
<td>HEXTORAW</td>
<td>INITCAP</td>
<td>INSTR</td>
</tr>
<tr>
<td>INSTRB</td>
<td>LEAST</td>
<td>LENGTH</td>
<td>LENGTHB</td>
<td>LN</td>
</tr>
<tr>
<td>LOG</td>
<td>LOWER</td>
<td>LPAD</td>
<td>LTRIM</td>
<td>MOD</td>
</tr>
<tr>
<td>NLSSORT</td>
<td>POWER</td>
<td>RAWTOHEX</td>
<td>REPLACE</td>
<td>ROUND</td>
</tr>
<tr>
<td>RPAD</td>
<td>RTRIM</td>
<td>SIGN</td>
<td>SIN</td>
<td>SINH</td>
</tr>
<tr>
<td>SQRT</td>
<td>STDDEV</td>
<td>SUBSTR</td>
<td>SUBSTRB</td>
<td>TAN</td>
</tr>
<tr>
<td>TANH</td>
<td>TRANSLATE</td>
<td>TRUNC</td>
<td>UPPER</td>
<td>VARIANCE</td>
</tr>
</tbody>
</table>

SQL Functions Which 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.
**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 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.

**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 listed in the table are general. Refer to documentation for your DRDA database for restrictions on datatype size and value limitations.

<table>
<thead>
<tr>
<th>DRDA Server</th>
<th>Oracle External</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR(N)</td>
<td>CHAR(N)</td>
<td>N &lt;= 255</td>
</tr>
<tr>
<td>VARCHAR (N)</td>
<td>VARCHAR2(N)</td>
<td>N &lt;= 2000</td>
</tr>
<tr>
<td></td>
<td>LONG</td>
<td>2000 &lt; N &lt;= 32740</td>
</tr>
<tr>
<td>LONG VARCHAR(N)</td>
<td>VARCHAR2(N)</td>
<td>N &lt;= 2000</td>
</tr>
<tr>
<td>LONG VARCHAR(N)</td>
<td>LONG</td>
<td>2000 &lt; N &lt;= 32740</td>
</tr>
<tr>
<td>CHAR(N) FOR BIT DATA</td>
<td>RAW(N)</td>
<td>N &lt;= 255</td>
</tr>
<tr>
<td>VARCHAR(N) FOR BIT DATA</td>
<td>RAW(N)</td>
<td>1 &lt;= N &lt;= 255</td>
</tr>
<tr>
<td>VARCHAR(N) FOR BIT DATA</td>
<td>LONG RAW(N)</td>
<td>255 &lt; N &lt;= 32740</td>
</tr>
</tbody>
</table>
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.

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.

Performing Graphic String Operations

The gateway does not support GRAPHIC or VARGRAPHIC datatypes.
Performing Date and Time Operations

The implementation of date and time data differs significantly in IBM DRDA databases and the Oracle server. The Oracle 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.

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.

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:

```sql
SELECT * FROM EMP WHERE HIREDATE = '03-MAR-81'
```

you must use:

```sql
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:

```sql
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:

```sql
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.
Performing Date Arithmetic

The following forms of SQL expression generally do not work correctly with the gateway:

\[ \text{date + number} \]
\[ \text{number + date} \]
\[ \text{date - number} \]
\[ \text{date1 - date2} \]

The date and number addition and subtraction \((\text{date + number, number + date, date - number})\) 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 \((\text{date1 - date2})\) gives results that vary by server.

---

**Note:** Avoid date arithmetic expressions in all gateway SQL until date arithmetic problems are resolved.
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 Oracle9i 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 Oracle9i 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.
HS_NLS_DATE_FORMAT Support

The following table lists the four patterns that can be used for the HS_NLS_DATE_FORMAT.

<table>
<thead>
<tr>
<th>DB2 Date Format</th>
<th>Pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR</td>
<td>DD.MM.YYYY</td>
<td>30.10.1994</td>
</tr>
<tr>
<td>ISO</td>
<td>YYYY-MM-DD</td>
<td>1994-10-30</td>
</tr>
<tr>
<td>JIS</td>
<td>YYYY-MM-DD</td>
<td>1994-10-30</td>
</tr>
<tr>
<td>USA</td>
<td>MM/DD/YYYY</td>
<td>10/30/1994</td>
</tr>
</tbody>
</table>

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';
```
DRDA Datatype to Oracle Datatype Conversion

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.

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.
Mapping the COUNT Function

The Oracle 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

Performing Zoned Decimal Operations

A zoned decimal field is described as packed decimal on an Oracle 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.
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.

This release of Oracle Transparent Gateway for IBM 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 14-38 for more information.

Using the `DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE` Function

To run a passthrough SQL statement using `DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE`, use the following syntax:

```sql
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.
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;
   /
Retrieving Results Sets Through Passthrough

Oracle Transparent Gateway for IBM DRDA provides a facility to retrieve results sets from a SELECT SQL statement entered through passthrough. Refer to Oracle9i Heterogeneous Connectivity Administrator's Guide for additional information.

Example

```sql
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;
END;/
```

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.
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 and DB2/400 catalog views. DB2/VM and DB2/UDB 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. All Oracle DB2/OS390 catalog views are supported by the Oracle Transparent Gateway for IBM DRDA. Refer to Appendix A for descriptions of Oracle DB2 catalog views. These views are completely compatible with the gateway.

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.

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.
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 IBM 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 15-4
- Passwords in the Gateway Initialization File on page 15-8
- Using the g4drpwd Utility on page 15-9
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.

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 server. When an application connects to the server, it supplies a user ID and password. The Oracle 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 Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris and Oracle Net documentation to determine the availability of this feature.

For more information about authenticating application logons, refer to the Oracle9i Administrator’s Reference.

**Defining and Controlling Database Links**

The information here is specific to the gateway. For additional information on database links, refer to the Oracle9i Administrator’s Reference.

**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.
TCP/IP Security

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.

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.

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 3-5, for a list of IBM documentation.
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.

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.SYSLUNAMES table
  
  The SYSIBM.SYSLUNAMES 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.

**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.
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.

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.
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.
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 "*". 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
ORADRDA
```

In the example above, the parameter "DRDA_RECOVERY_PASSWORD" is identified as requiring encryption. The user enters the value (in other words, "ORADRDA") 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 to new instances of Oracle Transparent Gateway for IBM 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 IBM DRDA, including the following sections:

- Migrating Existing V4 Gateway Instances to New Release on page 16-2
- Backout Considerations When Migrating to New Releases on page 16-3
- New and Changed Parameters When Migrating from Version 4 to Release 9i of the Gateway on page 16-3
- DRDA Server Considerations on page 16-6
- Oracle Net Considerations on page 16-6
Migrating Existing V4 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.

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".

Warning:

Do not install the Gateway over a previously existing Gateway installation. Doing so will corrupt that existing installation.

Step 2: Transferring init$SID.gtwboot Gateway Boot Initialization parameters.

In previous installations, two Gateway Initialization Files existed: init$SID.gtwboot and init$SID.ora. As mentioned in Chapter 11, "Configuring the Gateway", the parameters in the init$SID.gtwboot have been moved to a Startup Shell Script. Copy the parameters from the init$SID.gtwboot to the chosen Startup Shell Script and tailor them as needed. The format of the parameters can be found in the section in Appendix C, "DRDA-Specific Parameters".

Step 3: Transferring init$SID.ora Gateway Initialization File parameters.

Copy the init$SID.ora from the old Gateway instance to the new instance. The parameters in the init$SID.ora Gateway Initialization File have changed format. Refer to "Gateway Initialization File Parameters" on page C-3 in Appendix C, "DRDA-Specific Parameters".
Backout Considerations When Migrating to New Releases

During the migration from older version 4 gateway instances to the latest Oracle 9i 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 9i 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, you keep the old gateway home and instance configurations intact and operational in case there are problems with the upgrade. This will help ensure minimal downtime between changes to different gateway instances.

New and Changed Parameters When Migrating from Version 4 to Release 9i of the Gateway

This release of the Oracle Transparent Gateway for IBM DRDA introduces new and changed initialization parameters if you are migrating from a Version 4 gateway to Oracle 9i.

Note: The "New and Changed Parameters" section does not apply to you if you are migrating to Oracle9i from Version 8 of the Oracle Transparent Gateway for IBM DRDA.

New Parameters

The following section lists new parameters relevant to migration from Version 4 gateways.

New Startup Shell Parameters

The following new Startup Shell parameters have been introduced in this release of the gateway. They must appear in the Startup Shell Script:

- FDS_CLASS
- FDS_INSTANCE
New and Changed Parameters When Migrating from Version 4 to Release 9i of the Gateway

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_CAPABILITY
- DRDA_CMSRC_CM_IMMEDIATE
- DRDA_DESCRIBE_TABLE
- DRDA_LOCAL_NODE_NAME
- DRDA_READ_ONLY
- HS_RPC_FETCH_REBLOCKING
- HS_FDS_FETCH_ROWS
- HS_RPC_FETCH_SIZE
- HS_NLS_NCHAR

Note: The "HS_" parameters are specific to Oracle Heterogeneous Services. For details on HS parameters, refer to Oracle9i Heterogeneous Connectivity Administrator’s Guide.

Parameters That Have Been Changed in Usage

The following parameter’s usage has changed with this release of the gateway:

- DRDA_CONNECT_PARM
Parameters That Have Been Renamed

The following table presents a list of parameters have been renamed with this release of the gateway, and their corresponding old names. Refer to Oracle9i Heterogeneous Connectivity Administrator’s Guide for more detailed information about these parameters.

<table>
<thead>
<tr>
<th>New Name</th>
<th>Old Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS_COMMIT_STRENGTH_POINT</td>
<td>COMMIT_STRENGTH_POINT</td>
</tr>
<tr>
<td>HS_DB_DOMAIN</td>
<td>DB_DOMAIN</td>
</tr>
<tr>
<td>HS_DB_INTERNAL_NAME</td>
<td>DB_INTERNAL_NAME</td>
</tr>
<tr>
<td>HS_DB_NAME</td>
<td>DB_NAME</td>
</tr>
<tr>
<td>HS_DESCRIBE_CACHE_HWM</td>
<td>DESCRIBE_CACHE_HWM</td>
</tr>
<tr>
<td>HS_LANGUAGE</td>
<td>LANGUAGE</td>
</tr>
<tr>
<td>HS_NLS_DATE_FORMAT</td>
<td>NLS_DATE_FORMAT</td>
</tr>
<tr>
<td>HS_NLS_DATE_LANGUAGE</td>
<td>NLS_DATE_LANGUAGE</td>
</tr>
<tr>
<td>HS_OPEN_CURSORS</td>
<td>OPEN_CURSORS</td>
</tr>
<tr>
<td>HS_ROWID_CACHE_SIZE</td>
<td>ROWID_CACHE_SIZE</td>
</tr>
</tbody>
</table>

Obsolete Parameters

The following parameters are now obsolete. Please remove them from your configuration files:

- MODE
- SERVER_PATH
- ERROR_LOGGING
- ERROR_REPORTING
- ERRORTAG
- GATEWAY_SID
- GROUP_BY_OFF
- GTWDEBUG
- INCREMENT_CURSORS
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 (Release 9.2.0.1.0) is compatible with packages and views installed with the previous Version 8 and Version 4 releases. When migrating to the new release, you do not need to rebind existing packages or to re-install existing views in the DRDA server. However, this release of the gateway does have changes to the gateway tables and views which provide improved compatibility with Oracle server data dictionaries. It is recommended that you install the new tables and views as outlined in Chapter 11, "Configuring the Gateway". If you have modified certain DRDA parameters when migrating the gateway instance, then a rebind will be required of the package only. Appendix C lists the DRDA parameters which require a rebind if the value has been changed.

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 12-4 for detailed information.
This chapter provides information about error messages and error codes. This data is specific to this release of the Oracle Transparent Gateway for IBM DRDA, including the following sections:

- Interpreting Gateway Error Messages on page 17-2
- Mapped Errors on page 17-5
- Gateway Error Codes on page 17-6
- SQL Tracing and the Gateway on page 17-8
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 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

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 Oracle9i 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 Oracle9i Database Error Messages.
Errors Detected by the Gateway

Errors detected by the generic gateway are prefixed with "HGO-" and are documented in Oracle9i Database Error Messages.

An example error message is:

HGO-00706: HGO: Missing equal sign for parameter in initialization file.

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 17-6.
- **grc** specifies generic gateway errors detected in the DRDA layer. These errors are documented in the Oracle9i 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 DRDAREMOTE_NAME parameter in the **init<sid>.ora** file is not defined at the DRDA server:

ORA-28500: connection from ORACLE to a non-Oracle system returned the message:
TG4DRDA v9.2.0.1.0 grc=0, drc=-30061 (839C,0000), errp=GDJRFS2E
errmc=XNAME
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 initсид.ora file specifies a Side Information Profile that is not defined:

ORA-28501: communication error on heterogeneous database link
TG4DRDA v9.2.0.1.0 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.

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 v9.2.0.1.0 grc=0, drc=-30020 (839C,0000), errp=GDJMRCM
sqlcode=-805, sqlstate=51002, errd=FFFFFF9C,0,0,FFFFFFFC,0,0
errmc=124c
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>
<thead>
<tr>
<th>Description</th>
<th>sqlstate</th>
<th>Oracle Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>No rows selected</td>
<td>02000</td>
<td>0</td>
</tr>
<tr>
<td>Unique index constraint violated</td>
<td>23505</td>
<td>ORA-00001</td>
</tr>
<tr>
<td>Object does not exist</td>
<td>52004 or 42704</td>
<td>ORA-00942</td>
</tr>
<tr>
<td>Object name too long (more than 18 characters), and therefore object does not exist</td>
<td>54003 or 42622</td>
<td>ORA-00942</td>
</tr>
<tr>
<td>Insufficient privileges</td>
<td>42501</td>
<td>ORA-01031</td>
</tr>
<tr>
<td>Invalid CCSID (unimplemented character set conversion)</td>
<td>22522</td>
<td>ORA-01460</td>
</tr>
<tr>
<td>Invalid username/password; logon denied</td>
<td>N/A</td>
<td>ORA-01017</td>
</tr>
<tr>
<td>Divide by zero error</td>
<td>01519 or 01564</td>
<td>ORA-01476</td>
</tr>
</tbody>
</table>

The following is an example of a translated "object does not exist" error:

ORA-00942: table or view does not exist
TG4DRDA v9.2.0.1.0 grc=0, drc=-942 (893C,0001), errp=DSNXEDST
sqlcode=-204, sqlstate=52004, errd=32,0,0,FFFFFFF,0,0
errmc=AJONES.CXDCX
Gateway Error Codes

Listed below are the common Oracle Transparent Gateway for IBM 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 init$sid.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 init$sid.ora file is not supported by the Oracle Transparent Gateway for IBM 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 init$sid.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
   init$.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.
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.

SQL Tracing in the Oracle Database

The Oracle 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 Oracle9i 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.

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 g4dsvrd
```
This will create the debug gateway and store it in $ORACLE_HOME/bin/g4drsrvd. Next, change the Startup Shell Script to invoke the debug gateway. Our example assumes a SID of 'drdahoa1':

```bash
$ cd $ORACLE_HOME/bin
$ vi drdahoa1.sh

Find the line which says:
exec $ORACLE_HOME/bin/g4drsrv $*

and change it to:
exec $ORACLE_HOME/bin/g4drsrvd $*

Save the Startup Shell Script and then edit the Gateway Initialization File. 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:

```bash
# 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 string ‘*** HGAPARS ***’ and ‘*** HGAXMSQL ***’. The string after HGAPARS will be the incoming statement from the Oracle 9i 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, you should revert the ‘exec’ line in the Startup Shell Script to it’s previous form in order to use the production gateway again. You should also comment out the trace parameters in the Gateway Initialization Files.
This appendix includes the Oracle Transparent Gateway for IBM DRDA data dictionary views accessible to all users of an Oracle 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-4
Supported Views

The following is a list of Oracle data dictionary views which are supported by the gateway for DB2/OS390 and DB2/400 servers. This release of the gateway does not have data dictionary view support for DB2/VM and DB2/UDB DRDA servers.

- ALL_CATALOG
- ALL_COL_COMMENTS
- ALL_CONSTRAINTS
- ALL_CONS_COLUMNS
- 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
Supported Views

- USER_TABLES
- USER_TAB_COLUMNS
- USER_TAB_COMMENTS
- USER_USERS
- USER_VIEWS
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.

**ALL_CATALOG**

All tables, views, synonyms, and sequence accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of object</td>
</tr>
</tbody>
</table>

**ALL_COL_COMMENTS**

Comments on columns of accessible tables and views:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Object name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Comments on column</td>
</tr>
</tbody>
</table>
### ALL_CONS_COLUMNS

Information about accessible columns in constraint definitions:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>Name associated with the constraint definition</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name associated with table with constraint definition</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Name associated with column specified in the constraint definition</td>
</tr>
<tr>
<td>POSITION</td>
<td>Original position of column in definition</td>
</tr>
</tbody>
</table>
## ALL_CONSTRAINTS

Constraint definitions on accessible tables:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>Name associated with the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_TYPE</td>
<td>Type of constraint definition</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name associated with table with constraint definition</td>
</tr>
<tr>
<td>SEARCH_CONDITION</td>
<td>Text of search condition for table check</td>
</tr>
<tr>
<td>R_OWNER</td>
<td>Owner of table used in referential constraint</td>
</tr>
<tr>
<td>R_CONSTRAINT_NAME</td>
<td>Name of unique constraint definition for referenced table</td>
</tr>
<tr>
<td>DELETE_RULE</td>
<td>Delete rule for referential constraint</td>
</tr>
<tr>
<td>STATUS</td>
<td>Status of constraint</td>
</tr>
<tr>
<td>DEFERRABLE</td>
<td>Whether the constraint is deferrable</td>
</tr>
<tr>
<td>DEFERRED</td>
<td>Whether the constraint was initially deferred</td>
</tr>
<tr>
<td>VALIDATED</td>
<td>Whether all data obeys the constraint</td>
</tr>
<tr>
<td>GENERATED</td>
<td>Whether the name of the constraint is user or system generated</td>
</tr>
<tr>
<td>BAD</td>
<td>Constraint specifies a century in an ambiguous manner</td>
</tr>
<tr>
<td>RELY</td>
<td>Whether an enabled constraint is enforced or unenforced</td>
</tr>
<tr>
<td>LAST_CHANGE</td>
<td>When the constraint was las enabled or disabled</td>
</tr>
</tbody>
</table>
**ALL_INDEXES**

Description of indexes on tables accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the index</td>
</tr>
<tr>
<td>INDEX_NAME</td>
<td>Name of the index</td>
</tr>
<tr>
<td>INDEX_TYPE</td>
<td>Type of index</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the indexed object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the indexed object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of the indexed object</td>
</tr>
<tr>
<td>UNIQUENESS</td>
<td>Uniqueness status of the index</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the index</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_THRESHOLD</td>
<td>Threshold percentage of block space allowed per index entry</td>
</tr>
<tr>
<td>INCLUDE_COLUMN</td>
<td>Column ID of the last column to be included in index-organized table</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>Number of process freelists allocated to this segment</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>Number of freelist groups allocated to this segment</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING</td>
<td>Logging information</td>
</tr>
<tr>
<td>BLEVEL</td>
<td>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.</td>
</tr>
<tr>
<td>LEAF_BLOCKS</td>
<td>Number of leaf blocks in the index</td>
</tr>
</tbody>
</table>
### ALL_IND_COLUMNS

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTINCT_KEYS</td>
<td>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.</td>
</tr>
<tr>
<td>AVG_LEAF_BLOCKS_PER</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_DATA_BLOCKS_PER</td>
<td>N/A</td>
</tr>
<tr>
<td>CLUSTERING_FACTOR</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the index: VALID</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>Number of rows in the index</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>Size of the sample used to analyze the index</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>Date on which this index was most recently analyzed</td>
</tr>
<tr>
<td>DEGREE</td>
<td>Number of threads per instance for scanning the index</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>Number of instances across which the index is to be scanned</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>Whether this index is partitioned</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Whether the index is on a temporary table</td>
</tr>
<tr>
<td>GENERATED</td>
<td>Whether the name of the index is system generated</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>Whether the index is a secondary object</td>
</tr>
</tbody>
</table>

### column names

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX_OWNER</td>
<td>Owner of the index</td>
</tr>
<tr>
<td>INDEX_NAME</td>
<td>Name of the index</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the table or cluster</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the table or cluster</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name or attribute of object type column</td>
</tr>
<tr>
<td>COLUMN_POSITION</td>
<td>Position of column or attribute within the index</td>
</tr>
<tr>
<td>COLUMN_LENGTH</td>
<td>Indexed length of the column</td>
</tr>
</tbody>
</table>
### ALL_OBJECTS

Objects accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the object</td>
</tr>
<tr>
<td>OBJECT_NAME</td>
<td>Name of object</td>
</tr>
<tr>
<td>SUBOBJECT_NAME</td>
<td>Name of the subobject</td>
</tr>
<tr>
<td>OBJECT_ID</td>
<td>Object number of the object</td>
</tr>
<tr>
<td>DATA_OBJECT_ID</td>
<td>Dictionary object number of the segment that contains the object</td>
</tr>
<tr>
<td>OBJECT_TYPE</td>
<td>Type of object</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_DDL_TIME</td>
<td>N/A</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the object</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Whether the object is temporary</td>
</tr>
<tr>
<td>GENERATED</td>
<td>Was the name of this object system generated?</td>
</tr>
</tbody>
</table>

### ALL_SYNONYMS

All synonyms accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the synonym</td>
</tr>
<tr>
<td>SYNONYM_NAME</td>
<td>Name of the synonym</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the object referenced by the synonym</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object referenced by the synonym</td>
</tr>
<tr>
<td>DB_LINK</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### ALL_TABLES

Description of tables accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the table</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the table</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the table</td>
</tr>
<tr>
<td>CLUSTER_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>IOT_NAME</td>
<td>Name of the index organized table</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_USED</td>
<td>N/A</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>Number of process freelists allocated to this segment</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>Number of freelist groups allocated to this segment</td>
</tr>
<tr>
<td>LOGGING</td>
<td>Logging attribute</td>
</tr>
<tr>
<td>BACKED_UP</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>Number of rows in the table</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>EMPTY_BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_SPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>CHAIN_CNT</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_ROW_LEN</td>
<td>Average length of a row in the table in bytes</td>
</tr>
<tr>
<td>AVG_SPACE_FREELIST_BLOCKS</td>
<td>The average freespace of all blocks on a freelist</td>
</tr>
<tr>
<td>column name</td>
<td>description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>NUM_FREELIST_BLOCKS</td>
<td>The number of blocks on the freelist</td>
</tr>
<tr>
<td>DEGREE</td>
<td>The number of threads per instance for scanning the table</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>The number of instances across which the table is to be scanned</td>
</tr>
<tr>
<td>CACHE</td>
<td>Whether the cluster is to be cached in the buffer cache</td>
</tr>
<tr>
<td>TABLE_LOCK</td>
<td>Whether table locking is enabled or disabled</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>Sample size used in analyzing this table</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>Date on which this table was most recently analyzed</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>Indicates whether this table is partitioned</td>
</tr>
<tr>
<td>IOT_TYPE</td>
<td>If this is an index organized table</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Can the current session only see data that it placed in this object itself?</td>
</tr>
<tr>
<td>NESTED</td>
<td>If the table is a nested table</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>The default buffer pool for the object</td>
</tr>
</tbody>
</table>

**ALL_TAB_COLUMNS**

Columns of all tables, views, and clusters accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the table or view</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Table or view name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>Datatype of column</td>
</tr>
<tr>
<td>DATA_TYPE_MOD</td>
<td>Datatype modifier of the column</td>
</tr>
<tr>
<td>DATA_TYPE_OWNER</td>
<td>Owner of the datatype of the column</td>
</tr>
<tr>
<td>DATA_LENGTH</td>
<td>Maximum length of the column in bytes</td>
</tr>
<tr>
<td>DATA_PRECISION</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_SCALE</td>
<td>Digits to the right of decimal point in a number</td>
</tr>
</tbody>
</table>
### ALL_TAB_COMMENTS

Comments on tables and views accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of object</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Comments on the object</td>
</tr>
</tbody>
</table>
ALL_USERS

Information about all users of the database:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>Name of the user</td>
</tr>
<tr>
<td>USER_ID</td>
<td>N/A</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
</tbody>
</table>

ALL_VIEWS

Text of views accessible to the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the view</td>
</tr>
<tr>
<td>VIEW_NAME</td>
<td>Name of the view</td>
</tr>
<tr>
<td>TEXT_LENGTH</td>
<td>Length of the view text</td>
</tr>
<tr>
<td>TEXT</td>
<td>View text. Only the first row of text is returned, even if multiple rows exist.</td>
</tr>
<tr>
<td>TYPE_TEXT_LENGTH</td>
<td>Length of the type clause of the typed view</td>
</tr>
<tr>
<td>TYPE_TEXT</td>
<td>Type clause of the typed view</td>
</tr>
<tr>
<td>OID_TEXT_LENGTH</td>
<td>Length of the WITH OID clause of the typed view</td>
</tr>
<tr>
<td>OID_TEXT</td>
<td>WITH OID clause of the typed view</td>
</tr>
<tr>
<td>VIEW_TYPE_OWNER</td>
<td>Owner of the type of the view if the view is a typed view</td>
</tr>
<tr>
<td>VIEW_TYPE</td>
<td>Type of the view if the view is a typed view</td>
</tr>
</tbody>
</table>
COLUMN_PRIVILEGES

Grants on columns for which the user is the grantor, grantee, or owner, or PUBLIC is the grantee:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRANTEE</td>
<td>Name of the user to whom access was granted</td>
</tr>
<tr>
<td>OWNER</td>
<td>Username of the object’s owner</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Name of the column</td>
</tr>
<tr>
<td>GRANTOR</td>
<td>Name of the user who performed the grant</td>
</tr>
<tr>
<td>INSERT_PRIV</td>
<td>Permission to insert into the column</td>
</tr>
<tr>
<td>UPDATE_PRIV</td>
<td>Permission to update the column</td>
</tr>
<tr>
<td>REFERENCES_PRIV</td>
<td>Permission to reference the column</td>
</tr>
<tr>
<td>CREATED</td>
<td>Timestamp for the grant</td>
</tr>
</tbody>
</table>

DICTIONARY

List or data dictionary tables:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Table name</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Description of table</td>
</tr>
</tbody>
</table>

DUAL

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>A dummy column</td>
</tr>
</tbody>
</table>
TABLE_PRIVILEGES

Grants on objects for which the user is the grantor, grantee, or owner, or PUBLIC is the grantee:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRANTEE</td>
<td>Name of the user to whom access is granted</td>
</tr>
<tr>
<td>OWNER</td>
<td>Owner of the object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>GRANTOR</td>
<td>Name of the user who performed the grant</td>
</tr>
<tr>
<td>SELECT_PRIV</td>
<td>Permission to select from an object</td>
</tr>
<tr>
<td>INSERT_PRIV</td>
<td>Permission to insert into an object</td>
</tr>
<tr>
<td>DELETE_PRIV</td>
<td>Permission to delete from an object</td>
</tr>
<tr>
<td>UPDATE_PRIV</td>
<td>Permission to update an object</td>
</tr>
<tr>
<td>REFERENCES_PRIV</td>
<td>N/A</td>
</tr>
<tr>
<td>ALTER_PRIV</td>
<td>Permission to alter an object</td>
</tr>
<tr>
<td>INDEX_PRIV</td>
<td>Permission to create or drop an index on an object</td>
</tr>
<tr>
<td>CREATED</td>
<td>Timestamp for the grant</td>
</tr>
</tbody>
</table>

USER_CATALOG

Tables, views, synonyms, and sequences owned by the use:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of object</td>
</tr>
</tbody>
</table>
USER_COL_COMMENTS

Comments on columns of user’s tables and views:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Object name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Comments on column</td>
</tr>
</tbody>
</table>

USER_CONSTRAINTS

Constraint definitions on user’s tables:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>Name associated with the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_TYPE</td>
<td>Type of constraint definition</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name associated with table with constraint definition</td>
</tr>
<tr>
<td>SEARCH_CONDITION</td>
<td>Text of search condition for table check</td>
</tr>
<tr>
<td>R_OWNER</td>
<td>Owner of table used in referential constraint</td>
</tr>
<tr>
<td>R_CONSTRAINT_NAME</td>
<td>Name of unique constraint definition for referenced table</td>
</tr>
<tr>
<td>DELETE_RULE</td>
<td>Delete rule for referential constraint</td>
</tr>
<tr>
<td>STATUS</td>
<td>Status of constraint</td>
</tr>
<tr>
<td>DEFERRABLE</td>
<td>Whether the constraint is deferrable</td>
</tr>
<tr>
<td>DEFERRED</td>
<td>Whether the constraint was initially deferred</td>
</tr>
<tr>
<td>VALIDATED</td>
<td>Whether all data obeys the constraint</td>
</tr>
<tr>
<td>GENERATED</td>
<td>Whether the name of the constraint is user or system generated</td>
</tr>
<tr>
<td>BAD</td>
<td>Constraint specifies a century in an ambiguous manner</td>
</tr>
<tr>
<td>LAST_CHANGE</td>
<td>When the constraint was last enabled or disabled</td>
</tr>
</tbody>
</table>
USER_CONS_COLUMNS

Information about columns in constraint definitions owned by the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>Name associated with the constraint definition</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name associated with table with constraint definition</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Name associated with column specified in the constraint definition</td>
</tr>
<tr>
<td>POSITION</td>
<td>Original position of column in definition</td>
</tr>
</tbody>
</table>

USER_INDEXES

Description of the user’s own indexes:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX_NAME</td>
<td>Name of the index</td>
</tr>
<tr>
<td>INDEX_TYPE</td>
<td>Type of index</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the indexed object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the indexed object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of the indexed object</td>
</tr>
<tr>
<td>UNIQUENESS</td>
<td>Uniqueness status of the index</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the index</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>column name</td>
<td>description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PCT_THRESHOLD</td>
<td>Threshold percentage of block space allowed per index entry</td>
</tr>
<tr>
<td>INCLUDE_COLUMN</td>
<td>Column ID of the last column to be included in index-organized table</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>Number of process freelists allocated to this segment</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>Number of freelist groups allocated to this segment</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING</td>
<td>Logging information</td>
</tr>
<tr>
<td>BLEVEL</td>
<td>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.</td>
</tr>
<tr>
<td>LEAF_BLOCKS</td>
<td>Number of leaf blocks in the index</td>
</tr>
<tr>
<td>DISTINCT_KEYS</td>
<td>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.</td>
</tr>
<tr>
<td>AVG_LEAF_BLOCKS_PER</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_DATA_BLOCKS_PER</td>
<td>N/A</td>
</tr>
<tr>
<td>CLUSTERING_FACTOR</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the indexes: VALID</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>Number of rows in the index</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>Size of the sample used to analyze the index</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>Date on which this index was most recently analyzed</td>
</tr>
<tr>
<td>DEGREE</td>
<td>Number of threads per instance for scanning the index</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>Number of instances across which the index is to be scanned</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>Whether this index is partitioned</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Whether the index is on a temporary table</td>
</tr>
<tr>
<td>GENERATED</td>
<td>Whether the name of the index is system generated</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>Whether the index is a secondary object</td>
</tr>
</tbody>
</table>
USER_OBJECTS

Objects owned by the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT_NAME</td>
<td>Name of object</td>
</tr>
<tr>
<td>SUBOBJECT_NAME</td>
<td>Name of the subobject</td>
</tr>
<tr>
<td>OBJECT_ID</td>
<td>Object number of the object</td>
</tr>
<tr>
<td>DATA_OBJECT_ID</td>
<td>Dictionary object number of the segment that contains the object</td>
</tr>
<tr>
<td>OBJECT_TYPE</td>
<td>Type of object</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_DDL_TIME</td>
<td>N/A</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the object: VALID</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Whether the object is temporary</td>
</tr>
<tr>
<td>GENERATED</td>
<td>Was the name of this object system generated?</td>
</tr>
</tbody>
</table>

USER_SYNONYMS

The user’s private synonyms:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNONYM_NAME</td>
<td>Name of the synonym</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the object referenced by the synonym</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object referenced by the synonym</td>
</tr>
<tr>
<td>DB_LINK</td>
<td>N/A</td>
</tr>
</tbody>
</table>
USER_TABLES

Description of the user’s own tables:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Name of the table</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the table</td>
</tr>
<tr>
<td>CLUSTER_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>IOT_NAME</td>
<td>Name of the index organized table</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_USED</td>
<td>N/A</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>Number of process freelists allocated to this segment</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>Number of freelist groups allocated to this segment</td>
</tr>
<tr>
<td>LOGGING</td>
<td>Logging information</td>
</tr>
<tr>
<td>BACKED_UP</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>Number of rows in the table</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>EMPTY_BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_SPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>CHAIN_CNT</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_ROW_LEN</td>
<td>Average length of a row in the table in bytes</td>
</tr>
<tr>
<td>AVG_SPACE_FREELIST_BLOCKS</td>
<td>The average freespace of all blocks on a freelist</td>
</tr>
<tr>
<td>NUM_FREELIST_BLOCKS</td>
<td>The number of blocks on the freelist</td>
</tr>
<tr>
<td>column name</td>
<td>description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DEGREE</td>
<td>The number of threads per instance for scanning the table</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>The number of instances across which the table is to be scanned</td>
</tr>
<tr>
<td>CACHE</td>
<td>Whether the cluster is to be cached in the buffer cache</td>
</tr>
<tr>
<td>TABLE_LOCK</td>
<td>Whether table locking is enabled or disabled</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>Sample size used in analyzing this table</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>Date on which this table was most recently analyzed</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>Indicates whether this table is partitioned</td>
</tr>
<tr>
<td>IOT_TYPE</td>
<td>If this is an index organized table</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>Can the current session only see data that it placed in this object itself?</td>
</tr>
<tr>
<td>NESTED</td>
<td>If the table is a nested table</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>The default buffer pool for the object</td>
</tr>
</tbody>
</table>
**USER_TAB_COLUMNS**

Columns of user’s tables, views, and clusters:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Table, view, or cluster name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>Datatype of column</td>
</tr>
<tr>
<td>DATA_TYPE_MOD</td>
<td>Datatype modifier of the column</td>
</tr>
<tr>
<td>DATA_TYPE_OWNER</td>
<td>Owner of the datatype of the column</td>
</tr>
<tr>
<td>DATA_LENGTH</td>
<td>Maximum length of the column in bytes</td>
</tr>
<tr>
<td>DATA_PRECISION</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_SCALE</td>
<td>Digits to the right of decimal point in a number</td>
</tr>
<tr>
<td>NULLABLE</td>
<td>Does the column allow nulls? Value is n if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY key.</td>
</tr>
<tr>
<td>COLUMN_ID</td>
<td>Sequence number of the column as created</td>
</tr>
<tr>
<td>DEFAULT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_DEFAULT</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_DISTINCT</td>
<td>Number of distinct values in each column of the table</td>
</tr>
<tr>
<td>LOW_VALUE</td>
<td>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.</td>
</tr>
<tr>
<td>HIGH_VALUE</td>
<td>N/A</td>
</tr>
<tr>
<td>DENSITY</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_NULLS</td>
<td>The number of nulls in the column</td>
</tr>
<tr>
<td>NUM_BUCKETS</td>
<td>The number of buckets in histogram for the column</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>The date on which this column was most recently analyzed</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>The sample size used in analyzing this column</td>
</tr>
<tr>
<td>CHARACTER_SET_NAME</td>
<td>The name of the character set</td>
</tr>
<tr>
<td>CHAR_COL_DECL_LENGTH</td>
<td>The length of the character set</td>
</tr>
</tbody>
</table>
**USER_TAB_COMMENTS**

Comments on the tables and views owned by the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of object</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Comments on the object</td>
</tr>
</tbody>
</table>

**USER_USERS**

Information about the current user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>Name of the user</td>
</tr>
<tr>
<td>USER_ID</td>
<td>N/A</td>
</tr>
<tr>
<td>ACCOUNT_STATUS</td>
<td>Indicates if the account is locked, expired or unlocked</td>
</tr>
<tr>
<td>LOCK_DATE</td>
<td>Date the account was locked</td>
</tr>
<tr>
<td>EXPIRE_DATE</td>
<td>Date of expiration of the account</td>
</tr>
<tr>
<td>DEFAULT_TABLESPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>TEMPORARY_TABLESPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
<tr>
<td>EXTERNAL_NAME</td>
<td>User external name</td>
</tr>
</tbody>
</table>
### USER_VIEWS

Text of views owned by the user:

<table>
<thead>
<tr>
<th>column name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIEW_NAME</td>
<td>Name of the view</td>
</tr>
<tr>
<td>TEXT_LENGTH</td>
<td>Length of the view text</td>
</tr>
<tr>
<td>TEXT</td>
<td>First line of the view text</td>
</tr>
<tr>
<td>TYPE_TEXT_LENGTH</td>
<td>Length of the type clause of the typed view</td>
</tr>
<tr>
<td>TYPE_TEXT</td>
<td>Type clause of the typed view</td>
</tr>
<tr>
<td>OID_TEXT_LENGTH</td>
<td>Length of the WITH OID clause of the typed view</td>
</tr>
<tr>
<td>OID_TEXT</td>
<td>WITH OID clause of the typed view</td>
</tr>
<tr>
<td>VIEW_TYPE_OWNER</td>
<td>Owner of the type of the view if the view is a typed view</td>
</tr>
<tr>
<td>VIEW_TYPE</td>
<td>Type of the view if the view is a typed view</td>
</tr>
</tbody>
</table>
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 Startup Shell Script on page B-3
- Sample Oracle Net tnsnames.ora File on page B-3
- Sample Oracle Net listener.ora File on page B-4
Sample Gateway Initialization File

The following sample Gateway Initialization File (initdrdahoa1.ora) needs customization. For information on customizing this file, refer to "Configuring the Host" on page 11-4 in Chapter 11, "Configuring the Gateway". Also refer to Appendix C.

# HS specific parameters
#
#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=DRDACON1
DRDA_REMOTE_DB_NAME=DB2V2R3
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
Sample Startup Shell Script

The following is a sample Startup Shell Script (initdrdahoa1.sh). Refer to "Configuring the Host" on page 11-4 to determine if there is a need to customize the script. Also refer to the Startup Shell Script parameters in Appendix C for specific information about customizing them.

#!/bin/sh
#
# Oracle Transparent Gateway for IBM DRDA driver
#
FDS_CLASS='TG4DRDA_DB2MVS'; export FDS_CLASS
PATH="$ORACLE_HOME/tg4drda/admin":$PATH; export PATH
exec $ORACLE_HOME/bin/g4drsrv $*

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 12-4.

ipc-ora9=(DESCRIPTION=
    (ADDRESS=
        (PROTOCOL=IPC)
        (KEY=ORAIPC)
    )
    (CONNECT_DATA=(SID=ORA901))
)

ipc-gtw=(DESCRIPTION=
    (ADDRESS=
        (PROTOCOL=IPC)
        (KEY=ORAIPC)
    )
    (CONNECT_DATA=(SID=drdahoa1))
    (HS=)
)
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 12-4.

# 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=drdaho1a1)
   (ORACLE_HOME=/oracle/tg4drda/9.2.0)
   (PROGRAM=drdaho1a1.sh)
  )
 )

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 these lines under the LISTENER keyword:

  (ADDRESS=
    (PROTOCOL=IPC)
    (KEY=ORAIPC)
  )

with

  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=your_IP_node_name)
    (PORT=your_port_number)
  )

B-4 Oracle Transparent Gateway for IBM DRDA Installation and User's Guide
This appendix contains the DRDA-specific parameters defined in the Gateway Initialization File and the Startup Shell Script. 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-3
Modifying the Gateway Initialization File

If you change any parameters in the Gateway Initialization File, 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.

Setting Parameters in the Gateway Initialization File

Parameters specific to the gateway are stored in the Gateway Initialization File, \textit{init\text{sid}.ora}.

Syntax and Usage

Parameters and their values are specified according to the syntax rules specified by Heterogeneous Services. The general form is:

\[ [\text{set}] \ [\text{private}] \ \text{drda\_parameter} = \text{drda\_parameter\_value} \]

where:

- \text{drda\_parameter} is one of the DRDA parameters
- \text{drda\_parameter\_value} is a character string with contents dependent on the \text{drda\_parameter}.

The \text{set} and \text{private} keywords are optional and have the following effect. If the \text{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 \text{private} keyword is present, then the parameter and its value will not be uploaded to the Oracle server. In general, Oracle corporation recommends that the \text{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 \textit{Oracle9i Heterogeneous Connectivity Administrator’s Guide}.
Gateway Initialization File Parameters

Below is a list of gateway specific initialization file parameters and their descriptions. In addition to these parameters, additional generic Heterogeneous Services initialization file parameters may be set. Refer to the Oracle9i Heterogeneous Connectivity Administrator’s Guide for a list of additional parameters.

DRDA_CAPABILITY

Default value: none

Range of values: Refer to "Native Semantics" on page 14-24

Syntax: DRDA_CAPABILITY={FUNCTION/ON | OFF}, ...

DRDA_CAPABILITY specifies which Oracle mapped functions will be treated natively. In other words, no special pre/post processing will be done for these functions. They will be passed through to the DRDA server unmodified.

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.

DRDA_COMM_BUFLEN

Default value: 32767

Range of values: 512 through 32767

Syntax: DRDA_COMM_BUFLEN=num

DRDA_COMM_BUFLEN specifies the communications buffer length. This is a number indicating the size of the SNA send/receive buffer in bytes.
Gateway Initialization File Parameters

**DRDA_CONNECT_PARM (SNA format)**

**Default value:** DRDACON1

**Range of values:** any alphanumeric string 1 to 8 characters in length

**Syntax:** `RDA_CONNECT_PARM=name`

DRDA_CONNECT_PARM specifies the Side Information name. Refer to Chapter 6, Chapter 7, Chapter 8, and Chapter 9 for details.

**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.

**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.
Gateway Initialization File Parameters

**DRDA_DEFAULT_CCSID**

Default value: none

Range of values: any supported DRDA server CCSID

Syntax: \texttt{DRDA\_DEFAULT\_CCSID}\texttt{=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).

**DRDA_DESCRIBE_TABLE**

Default value: TRUE

Range of values: \{TRUE|FALSE\}

Syntax: \texttt{DRDA\_DESCRIBE\_TABLE}\texttt{=(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 please disable this option.
**DRDA_DISABLE_CALL**

Default value: TRUE  

Range of values: {TRUE | FALSE}  

Syntax: \texttt{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 above and DB2/400 Version 3.1 and above. 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.

---

**DRDA_FLUSH_CACHE**

Default value: SESSION  

Range of values: {SESSION | COMMIT}  

Syntax: \texttt{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.
**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 defined to the package when it is created. All SQL statements 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>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHG</td>
<td>Change (default for DB2/400)</td>
</tr>
<tr>
<td>CS</td>
<td>Cursor Stability (default for DB2/UDB, DB2/OS390, and DB2/VM)</td>
</tr>
<tr>
<td>RR</td>
<td>Repeatable Read</td>
</tr>
</tbody>
</table>

*Rebinding Required*: Any change to this parameter requires you to rebind.

---

**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.
**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 13-6 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.

**DRDA_OVERRIDE_FROM_CODEPAGE (obsolete)**

**Default value:** none

**Range of values:** any valid CCSID

**Syntax:** DRDA_OVERRIDE_FROM_CODEPAGE=ccsid

DRDA_OVERRIDE_FROM_CODEPAGE, when used together with DRDA_OVERRIDE_TO_CODEPAGE, provides a method of forcing a certain character set codepage or CCSID to be substituted for an unsupported codepage within the DRDA server database. This parameter should be used only if you get one of the following errors when using the gateway:

ORA-28527 : Heterogeneous Services datatype mapping error

DRDA: drc=865B,gdc=-332,inf=0001,sqlcode=0,sqlstate=57017,errp=GDJREACS

DRDA: errmc=xxx850

DRDA: errd=865B,0,0,0,0

where xxx is the unsupported CCSID.

Set DRDA_OVERRIDE_FROM_CODEPAGE to the unsupported codepage (xxx), and set DRDA_OVERRIDE_TO_CODEPAGE to a compatible codepage. Use caution when setting these parameters because they cause an incorrect codepage to be used for character translation, and certain characters in the data will be translated incorrectly. Refer to Appendix D, "National Language Support", for information on supported codepages.

This parameter is deprecated and will be removed in a future release of the gateway. Use the Codepage Map Facility.
DRDA_OVERRIDE_TO_CODEPAGE (obsolete)

Default value: none

Range of values: Any valid CCSID

Syntax: DRDA_OVERRIDE_TO_CODEPAGE=ccsid

DRDA_OVERRIDE_TO_CODEPAGE, when used together with DRDA_OVERRIDE_FROM_CODEPAGE, provides a method of forcing a certain character set codepage or CCSID to be substituted for an unsupported codepage within the DRDA server database. Set DRDA_OVERRIDE_TO_CODEPAGE to a compatible and supported codepage. For more information, refer to the section, "DRDA_OVERRIDE_FROM_CODEPAGE (obsolete)" on page C-8. Refer to Appendix D, "National Language Support", for information on supported codepages.

This parameter is deprecated and will be removed in a future release of the gateway. Use the Codepage Map Facility.

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.
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.

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.
**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.

**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.

**DRDA_RDBMS_TYPE (obsolete)**

This parameter has become obsolete. Refer to parameters "FDS_CLASS" and "FDS_INSTANCE" later in this chapter.
Gateway Initialization File Parameters

**DRDA_READ_ONLY**

Default value: FALSE

Range of values: {TRUE | FALSE}

Syntax: \texttt{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.

**DRDA_RECOVERY_PASSWORD**

Default value: none

Range of values: any valid password

Syntax: \texttt{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 13-8. Also refer to Chapter 15 for information about security and about encrypting passwords.

**DRDA_RECOVERY_USERID**

Default value: ORARECOV

Range of values: any valid user ID

Syntax: \texttt{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 13-8.
**Gateway Initialization File Parameters**

**DRDA-Specific Parameters**

---

**DRDA_REMOTE_DB_NAME**

Default value:  DB2V2R3  
Range of values: an alphanumeric string 1 to 18 characters in length  
Syntax:  

DRDA_REMOTE_DB_NAME specifies the DRDA server location name. This is an identifying name assigned to the server for DRDA purposes. A technique for determining this name using a SQL SELECT statement is discussed in each of the server-specific installation sections in Chapter 5, "Configuring the DRDA Server".

---

**DRDA_SECURITY_TYPE**

Default value: PROGRAM  
Range of values: \{PROGRAM | SAME\}  
Syntax:  

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.
Gateway Initialization File Parameters

FDS_CLASS

Default value: TG4DRDA_DB2MVS

Range of values: Refer to the table below for a list of valid values

Syntax: FDS_CLASS='TG4DRDA_DB2MVS' ; export FDS_CLASS

FDS_CLASS specifies the capability classification used by Oracle 9i and the gateway. These values might change from release to release, depending on whether the gateway capabilities change.

The valid values for FDS_CLASS are as follows:

Default:

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

FDS_CLASS_VERSION

Default value: 1

Range of values: 1

Syntax: FDS_CLASS_VERSION=1; export FDS_CLASS_VERSION

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.
**FDS_INSTANCE**

**Default value:**  DRD1  
**Range of values:**  the name of the gateway SID  
**Syntax:**  FDS_INSTANCE='drdahoa1' ; export FDS_INSTANCE  
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 defined in the TNS Listener entry.

**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 server. This parameter will be effected 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 Server" in the *Oracle9i Heterogeneous Connectivity Administrator's Guide*. 
LOG_DESTINATION

Default value: $ORACLE_HOME/tg4drda/log/gateway sid_pid.log
Range of values: any valid file path
Syntax: LOG_DESTINATION=logpath; export LOG_DESTINATION

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.

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.

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.
**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.

**TRACE_LEVEL**

**Default Value:** 0  
**Range of values:** 0-255  
**Syntax:** TRACE_LEVEL = number; export TRACE_LEVEL

TRACE_LEVEL specifies a code tracing level. This value determines the level of detail which is logged to the gateway’s logfile during execution. This parameter is primarily used for diagnostics.
This appendix documents the National Language Support (NLS) information for the Oracle Transparent Gateway for IBM DRDA. This supplements the general Oracle NLS information found in the *Oracle9i Application Developer’s Guide*.

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, Oracle9i 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-5
- Gateway Configuration on page D-6
- Gateway Codepage Map Facility on page D-9
- Message Availability on page D-11
- Example of NLS Configuration on page D-12
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.

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.
### Table D-1 Parameters Needed for NLS Processing in Your System's Environments

<table>
<thead>
<tr>
<th>Environment</th>
<th>Parameter or Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>NLS_LANG</td>
<td>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. Refer to “Client and Oracle Integrating Server Configuration” on page D-4 for detailed information.</td>
</tr>
<tr>
<td>Oracle integrating server</td>
<td>CHARACTER SET</td>
<td>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. Refer to “Client and Oracle Integrating Server Configuration” on page D-4 for detailed information.</td>
</tr>
<tr>
<td>Oracle Transparent Gateway for IBM DRDA</td>
<td>ORA_NLS33</td>
<td>An environmental variable. ORA_NLS33 determines where the gateway loads its character sets and other language data. Refer to “Gateway Language Interaction with DRDA Server” on page D-5 for detailed information.</td>
</tr>
<tr>
<td>Oracle Transparent Gateway for IBM DRDA</td>
<td>NLS_LANG</td>
<td>An environmental variable. NLS_LANG defines the character set used for communication between the gateway and the Oracle integrating server. Refer to “Gateway Language Interaction with DRDA Server” on page D-5 for detailed information.</td>
</tr>
<tr>
<td>Oracle Transparent Gateway for IBM DRDA</td>
<td>HS_LANGUAGE</td>
<td>An initialization parameter. HS_LANGUAGE defines the character set used for communication between the gateway and the DRDA server. Refer to “Gateway Language Interaction with DRDA Server” on page D-5 for detailed information.</td>
</tr>
</tbody>
</table>
Client and Oracle Integrating Server Configuration

There are a number of NLS parameters that control NLS processing between the Oracle 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 *Oracle9i Database Administrator’s Guide*. These parameters do not directly affect gateway processing. However, you must ensure that the character set (specified by the Oracle server NLS_LANG environment variable) is compatible with the character sets you specify on the gateway and DRDA server.

When you create your Oracle database, the character set used to store data is specified by the CHARACTER SET parameter. 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 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 Oracle9i Database Administrator’s Guide.

Note that this does not mean that the gateway character set must be the same as the Oracle server character set. The Oracle Net facility will be performing implicit conversion between the Oracle server character set and the gateway character set.

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 it will be conversing in. 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 it will be conversing in. This will be the CCSID that the DB2 database was generated with. 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.SYSTRINGS table (this table has different names for the various DB2 versions). It is possible to add additional character set mappings to this table using DB2 utilities. Please refer to the DB2 Installation documentation for details.

The setting of the HS_LANGUAGE parameter in the gateways initSid.ora determines which CCSID is used by the Gateway for the conversation. Refer to "Gateway Codepage Map Facility" on page D-9 for the list of supported ASCII based Oracle Character Sets mapped to CCSIDs.

Note again that it is not necessary for the gateway character set to be the same as the Oracle server character set. In many cases, it is not feasible to set the gateway character set equal to the Oracle server character set because the DRDA server will not have a valid translation for it. Instead, choose a character set which 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 server character set.
Gateway Configuration

After the gateway is installed, there are several parameters you must change in order to customize for NLS support.

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_DATE_FORMAT**
- **HS_NLS_DATE_LANGUAGE**

**HS_LANGUAGE**

`HS_LANGUAGE` defines the character set 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:

\[ HS\_LANGUAGE=\text{language[\_territory.character\_set]} \]

where:

- \text{language} can be any valid language.
- \text{territory} is optional, and defaults to AMERICA.
- \text{character\_set} is optional and defaults to WE8ISO8859P1. The supported character sets are: US7ASCII, US8PC437, WE8PC850, WE8MSWIN1252, and WE8ISO8859P1, IW8ISO8859P8.

If you omit the HS_LANGUAGE parameter from \textit{init\_sid.ora}, then the default setting is AMERICAN\_AMERICA.US7ASCII. EBCDIC character sets are not supported. The values for \text{language} and \text{territory} (such as AMERICAN\_AMERICA) must be valid, but they have no effect on translations.

**HS_NLS_DATE_FORMAT**

\textsc{HS\_NLS\_DATE\_FORMAT} specifies the format for dates used by the DRDA server.

The syntax of the NLS_DATE_FORMAT parameter is:

\[ HS\_NLS\_DATE\_FORMAT=\text{date\_format} \]

where \text{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.

**HS_NLS_DATE_LANGUAGE**

\textsc{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.
NLS Parameters in the Gateway Startup Shell Script

Two parameters that affect NLS support are in the gateway Startup Shell Script:

- ORA_NLS33
- NLS_LANG

**ORA_NLS33**

ORA_NLS33 specifies the directory where the gateway loads its character sets and other language data, `$ORACLE_HOME/ocommon/nls/admin/data`.

The ORA_NLS33 parameter must be specified as follows:

```
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data; EXPORT ORA_NLS33
```

If you omit the ORA_NLS33 parameter, then the language, territory, and character set default to `AMERICAN_AMERICA.US7ASCII`, even if you have set NLS_LANG to some other value.

**NLS_LANG**

NLS_LANG defines the character set that is used for communication between the gateway and the Oracle server. The syntax of the NLS_LANG parameter is:

```
NLS_LANG= language [_ territory.character_set]; EXPORT NLS_LANG
```

where:

- `language` must match the language specified in the HS_LANGUAGE parameter of `initsid.ora`.
- `territory` must match the language specified in the HS_LANGUAGE parameter of `initsid.ora`.

- `character_set` is optional, and defaults to `AMERICA`. If you specify `territory`, then it must match the territory specified in the HS_LANGUAGE parameter of `initsid.ora`.

If you omit the NLS_LANG parameter, then the default setting is `AMERICAN_AMERICA.US7ASCII`. 
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 syntax of the map is: `CCSID direction Oracle_CharacterSet`

where:

- `CCSID` is the IBM coded character set identifier
- `direction` is one of the following:
  - `=` means mapping is bi-directional
  - `<` 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.

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 IBM DRDA:

```plaintext
# Copyright (c) Oracle Corporation 2001. All rights reserved.
# Transparent Gateway for IBM DRDA - CodePage/Oracle CharacterSet Map
37 = WE8EBCDIC37  # United States/Canada
273 = D8EBCDIC273  # Austria/Germany
277 = DK8EBCDIC277  # Denmark/Norway
278 = S8EBCDIC278  # Finland/Sweden
280 = I8EBCDIC280  # Italy
284 = WE8EBCDIC284  # Latin America/Spain
285 = WE8EBCDIC285  # United Kingdom
297 = F8EBCDIC297  # France
#420 = AR8EBCDICX  # Arabic Bilingual (USA English)
420 = AR8XBASIC  # Arabic Bilingual (USA English)
424 = IW8EBCDIC424  # Israel (Hebrew)
437 = US8PC437  # Personal Computer, USA
500 = WE8EBCDIC500  # International
813 = EL8ISO8859P7  # Greek
819 = WE8ISO8859P1  # ISO/ANSI Multilingual
838 = TH8TISEBCDIC  # Thai w/Low-Tone Marks & Ancient Chars
850 < US7ASCII  # Multilingual Page - Personal Computer
850 = WE8PC850  # Multilingual Page - Personal Computer
864 = AR8ISO8859P6  # Arabic - Personal Computer
870 = EE8EBCDIC870  # Latin 2, Multilingual/ROECE
871 = WE8EBCDIC871  # Iceland - CECP
875 = EL8EBCDIC875  # Greece
904 > US7ASCII  # Republic of China (ROC) - Personal Computer
912 = EE8ISO8859P2  # Latin 2 8-bit
916 = IW8ISO8859P8  # Israel (Hebrew)
1025 = CL8EBCDIC1025  # Cyrillic, Multiling
1086 = IWE8EBCDIC1086  # Israel
1252 = WE8MSWIN1252  # Latin 1 - MS-Windows
1253 = EL8MSWIN1253  # Greek - MS-Windows
#```

D-10 Oracle Transparent Gateway for IBM DRDA Installation and User's Guide
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.

### Message Availability

Whether a language message module is available depends on which modules are installed in the Oracle product set 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.
Example of NLS Configuration

Following is an example of all the settings needed to configure the gateway, DRDA server, Oracle 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 init*sid*.ora file:**

```plaintext
HS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P1
HS_NLS_DATE_FORMAT=YYYY-MM-DD
```

**DRDA server CCSID:**

273 (DBEBCDIC273)

**Oracle server and client setting for database:**

```sql
SELECT USERENV('language') FROM DUAL;
USERENV('LANGUAGE')
----------------------------
AMERICAN_AMERICA.WE8ISO8859P1
```

**Oracle server and client environment variables:**

```plaintext
NLS_LANG=GERMAN_GERMANY.WE8ISO8859P1
```
Configuration Worksheet

The table below is a worksheet that lists all of the parameter names and the reasons you will need them for configuring the Gateway and the Communications Interfaces (SNA and TCP/IP). Use the worksheet to gather the specific information you need before beginning the configuration process.

Table E–1  List of Parameters Needed to Configure the Gateway

<table>
<thead>
<tr>
<th>Reason</th>
<th>Name of Parameter Needed</th>
<th>Your Specific Parameters Here</th>
</tr>
</thead>
<tbody>
<tr>
<td>For: Gateway’s Oracle Home</td>
<td>ORACLE_HOME</td>
<td>___________________________________________</td>
</tr>
<tr>
<td>For: Gateway’s System ID</td>
<td>ORACLE_SID</td>
<td>___________________________________________</td>
</tr>
<tr>
<td>For: Primary Service Definition</td>
<td>SNA Network Name</td>
<td>___________________________________________</td>
</tr>
<tr>
<td>For: Primary Service Definition</td>
<td>Control Point Name</td>
<td>___________________________________________</td>
</tr>
<tr>
<td>For: Connection Properties Address</td>
<td>Remote Network Address</td>
<td>___________________________________________</td>
</tr>
<tr>
<td>For: Connection Properties Address</td>
<td>Remote SAP Address</td>
<td>___________________________________________</td>
</tr>
<tr>
<td>For: System Identification: Local Node Name and Remote Node Name</td>
<td>For Each: Network Name</td>
<td>___________________________________________</td>
</tr>
</tbody>
</table>
### Table E–1 List of Parameters Needed to Configure the Gateway

<table>
<thead>
<tr>
<th>Reason</th>
<th>Name of Parameter Needed</th>
<th>Your Specific Parameters Here</th>
</tr>
</thead>
<tbody>
<tr>
<td>For: System Identification: Local Node Name and Remote Node Name</td>
<td>Control Point Name</td>
<td></td>
</tr>
<tr>
<td>For: System Identification: Local Node Name and Remote Node Name</td>
<td>Local Node ID</td>
<td></td>
</tr>
<tr>
<td>For: System Identification: Local Node Name and Remote Node Name</td>
<td>Remote Node ID</td>
<td></td>
</tr>
<tr>
<td>For: Creating a Local LU Definition:</td>
<td>LU Alias</td>
<td></td>
</tr>
<tr>
<td>For: Creating a Local LU Definition:</td>
<td>LU Name</td>
<td></td>
</tr>
<tr>
<td>For: General APPC Mode Definition</td>
<td>Mode Name</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Limits</td>
<td>Parallel Session Limit</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Limits</td>
<td>Minimum Contention Winner Limit</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Limits</td>
<td>Partner Min Contention Winner Limit</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Limits</td>
<td>Automatic Activation Limit</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Characteristics</td>
<td>Pacing Send Count</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Characteristics</td>
<td>Pacing Receive Count</td>
<td></td>
</tr>
<tr>
<td>For: APPC Mode Characteristics</td>
<td>Max Send RU Size</td>
<td></td>
</tr>
<tr>
<td>Reason</td>
<td>Name of Parameter Needed</td>
<td>Your Specific Parameters Here</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>For: Remote LU Definition, General Properties</td>
<td>Appropriate Connection name</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Remote LU Definition, General Properties</td>
<td>LU Alias</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Remote LU Definition, General Properties</td>
<td>Network Name</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Remote LU Definition, General Properties</td>
<td>Uninterpreted Network Name</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Remote LU Properties Options</td>
<td>Any Security Options needed</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Remote collection ID</td>
<td>DRDA_PACKAGE_COLLID</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Remote package name</td>
<td>DRDA_PACKAGE_NAME</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Creating CPI-C Symbolic Destination Names (Side Information Profiles), general information</td>
<td>Appropriate Name for each Side Information Profile</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Creating CPI-C Symbolic Destination Names (Side Information Profiles), general information</td>
<td>Appropriate Mode</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Partner Information in CPI-C Name Properties</td>
<td>TP Name</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Partner Information in CPI-C Name Properties</td>
<td>Partner LU Name Alias</td>
<td>____________________________</td>
</tr>
<tr>
<td>For: Configuring TCP/IP</td>
<td>Local Hostname, Domain Name</td>
<td>____________________________</td>
</tr>
</tbody>
</table>
Table E–1  List of Parameters Needed to Configure the Gateway

<table>
<thead>
<tr>
<th>Reason</th>
<th>Name of Parameter Needed</th>
<th>Your Specific Parameters Here</th>
</tr>
</thead>
<tbody>
<tr>
<td>For: Configuring TCP/IP</td>
<td>IP Address</td>
<td></td>
</tr>
<tr>
<td>For: Configuring TCP/IP</td>
<td>Network Mask</td>
<td></td>
</tr>
<tr>
<td>For: Configuring TCP/IP</td>
<td>Name Server IP Address</td>
<td></td>
</tr>
<tr>
<td>For: Configuring TCP/IP</td>
<td>Destination Hostname or IP Address</td>
<td></td>
</tr>
<tr>
<td>For: Configuring TCP/IP</td>
<td>Destination Service Port Number</td>
<td></td>
</tr>
<tr>
<td>For: Recovery user ID</td>
<td>DRDA_RECOVERY_USERID</td>
<td></td>
</tr>
<tr>
<td>For: Recovery Password</td>
<td>DRDA_RECOVERY_PASSWORD</td>
<td></td>
</tr>
<tr>
<td>For: Remote Database Name</td>
<td>DRDA_REMOTE_DB_NAME</td>
<td></td>
</tr>
<tr>
<td>For: Connection Parameter</td>
<td>DRDA_CONNECT_PARM</td>
<td></td>
</tr>
<tr>
<td>For: Owner ID of DRDA package</td>
<td>DRDA_PACKAGE_OWNER</td>
<td></td>
</tr>
<tr>
<td>For: DB Name used with Oracle server</td>
<td>HS_DB_NAME</td>
<td></td>
</tr>
<tr>
<td>For: DB Domain used with Oracle server</td>
<td>HS_DB_DOMAIN</td>
<td></td>
</tr>
</tbody>
</table>
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.
The following table lists the Oracle SQL functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>MOD</td>
</tr>
<tr>
<td>ACOS</td>
<td>MONTHS_BETWEEN</td>
</tr>
<tr>
<td>ADD_MONTHS</td>
<td>NEW_TIME</td>
</tr>
<tr>
<td>ASCII</td>
<td>NEXT_DAY</td>
</tr>
<tr>
<td>ASIN</td>
<td>NLS_INITCAP</td>
</tr>
<tr>
<td>ATAN</td>
<td>NLS_LOWER</td>
</tr>
<tr>
<td>ATAN2</td>
<td>NLS_UPPER</td>
</tr>
<tr>
<td>CEIL</td>
<td>NLSSORT</td>
</tr>
<tr>
<td>CHAR_TO_ROWID</td>
<td>POWER</td>
</tr>
<tr>
<td>CHR</td>
<td>RAWTOHEX</td>
</tr>
<tr>
<td>CONVERT</td>
<td>REPLACE</td>
</tr>
<tr>
<td>COS</td>
<td>ROUND</td>
</tr>
<tr>
<td>COSH</td>
<td>ROWIDTOCHAR</td>
</tr>
<tr>
<td>DECODE</td>
<td>RPAD</td>
</tr>
<tr>
<td>DUMP</td>
<td>RTRIM</td>
</tr>
<tr>
<td>EXP</td>
<td>SIGN</td>
</tr>
<tr>
<td>FLOOR</td>
<td>SIN</td>
</tr>
<tr>
<td>GREATEST</td>
<td>SINH</td>
</tr>
<tr>
<td>HEXTORAW</td>
<td>SOUNDEX</td>
</tr>
<tr>
<td>Function</td>
<td>Result</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>INITCAP</td>
<td>SQRT</td>
</tr>
<tr>
<td>INSTR</td>
<td>STDDEV</td>
</tr>
<tr>
<td>INSTRB</td>
<td>SUBSTR</td>
</tr>
<tr>
<td>LAST_DAY</td>
<td>SUBSTRB</td>
</tr>
<tr>
<td>LEAST</td>
<td>SYSDATE</td>
</tr>
<tr>
<td>LENGTH</td>
<td>TAN</td>
</tr>
<tr>
<td>LENGTHB</td>
<td>TANH</td>
</tr>
<tr>
<td>LN</td>
<td>TO_CHAR</td>
</tr>
<tr>
<td>LOG</td>
<td>TO_DATE</td>
</tr>
<tr>
<td>LOWER</td>
<td>TO_LABEL</td>
</tr>
<tr>
<td>LPAD</td>
<td>TO_MULTI_BYTE</td>
</tr>
<tr>
<td>LTRIM</td>
<td>TO_NUMBER</td>
</tr>
<tr>
<td>TO_SINGLE_BYTE</td>
<td>USER</td>
</tr>
<tr>
<td>TRANSLATE</td>
<td>USERENV</td>
</tr>
<tr>
<td>TRUNC</td>
<td>VARIANCE</td>
</tr>
<tr>
<td>UID</td>
<td>VSIZE</td>
</tr>
<tr>
<td>UPPER</td>
<td></td>
</tr>
</tbody>
</table>
This appendix contains sample applications that can be used with the gateway:

- **DB2INS** on page G-1
- **ORAIND** on page G-4
DB2INS

DB2INS is a sample DB2 stored procedure that inserts a row into a DB2 table. This procedure uses the SIMPLE linkage convention.

```c
/* 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>
EXEC SQL INCLUDE SQLCA;
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;
```
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;
}
ORAIND

ORAIND is a sample host program that calls a DB2 stored procedure (DB2INS) to insert a row into a DB2 table.

`#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()
{
    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;

    EXEC SQL CONNECT :username IDENTIFIED BY :password;
    printf("Connected to ORACLE as user: %s\n", username.arr);
    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); */
/* */
glyphicon-highlight envision-emoji-defining-gateway
EXEC SQL ROLLBACK RELEASE;
exit(1);
}
Symbols

$DISPLAY
  environmental variables, 7-4, 9-4
$ORACLE_HOME, 7-4, 9-4

A

ABS SQL function, 14-25, F-1
accessing
  DRDA servers, 11-14
gateway, 11-13
  main topic, 13-4
ACOS function, F-1
ADD_MONTHS function, F-1
Advanced Program to Program Communication (APPC/LU6.2) protocol, 6-1, 7-1, 8-1, 9-1
Advanced Security
  CHECKSUM command, 12-6
  encryption, 12-6
  international version types supported, 12-7
  resetting configuration parameters on gateway, 12-8
  setting test parameters for gateway, 12-7
  setting test parameters for Oracle integrating server, 12-7
  setting up for test, 12-7
  testing gateway and Oracle integrating server, 12-8
export encryption algorithms, 12-6
function on the gateway, 1-6
purpose, 1-6
test error, 12-8
12660, 12-7
AGW ADD USERID command, 15-6
AGW DELETE USERID command, 15-6
AIX
  AIX_RS6K, default value, C-7
  System Management Interface Tool, 8-3
  user ID for DB2/UDB, 5-7
alias
  CPI-C destination, 7-15, 9-16
  DB2, C-5
  Ethernet parameters, 9-11
  local LU alias, 7-10
  partner LU, 8-5
alias objects, DB2
  known restrictions, 2-6
ALL_CATALOG view, A-4
ALL_COL_COMMENTS view, A-4
ALL_CON_COLUMNS view, A-5
ALL_CONSTRAINTS view, A-6
ALL_DB_LINKS data dictionary view, 13-3
ALL_INDEXES view, A-7
ALL_OBJECTS view, A-9
ALL_SYNONYMS view, A-9
ALL_TAB_COMMENTS view, A-12
ALL_TABLES view, A-10
ALL_USERS view, A-13
ALL_VIEWS view, A-13
allocation mode, SNA session,
  DRDA_CMSRC_CM_IMMEDIATE, C-4
ALTER session statement, 13-3
ANSI-standard SQL, 1-7, 1-16
API (application program interface), 12-3
APPC
  concurrent connections, 3-2
  configuring IBM Communication Server, 8-1
configuring SNAP-IX interfaces, 7-1
configuring SUNLINK, 6-1
conversations, 8-3
creating SNAP-IX profiles, 7-2
creating SNAPplus2 profiles, 9-2
DB2/VM, 15-6
number of database link connections, 14-8
password length, 17-7
user ID length, 17-6
APPC VTAM Support (AVS)
also see AVS, 15-6
known restrictions, 2-6
APPC/LU6.2
parallel session limits, 8-4
protocol, 1-4, 7-1, 9-1
side information profile, 8-5
APPEND command
supported by COPY, 13-10
application
authenticating logons, 15-2
portability, 1-17
server support, 1-6
application development on the gateway, 1-19
application program interface
see API
APPN
node, advanced peer-to-peer networking
node, 8-5
support requires VTAM Version 4, 8-5
architecture of the gateway, 1-11
array size, 1-15
how determined, 14-3
AS/400
command
DSPRDBDIRE, 5-6
communications
configuring, 5-5
files and file members
accessing, 13-5
library name, DRDA_PACKAGE_COLLID, C-9
ASCII
code page, D-5
function, F-1
sort order, 14-25
tables, known restrictions, 2-3
translated from EBCDIC, 14-27
US7ASCII, D-7
US7ASCII as default, D-4
ASCII SQL function, 14-25
ASIN function, F-1
ATAN function, F-1
ATAN2 function, F-1
authority
CONNECT, 15-6
eexecute, 15-6
autonomy, site, 1-9
available products, 4-6
available products menu, 4-6
AVS
also see APPC VTAM Support, 2-6
configuring, 5-9
DB2/VM, 15-6
mapping user IDs, 2-6, 15-6
mapping user IDs (DB2/VM)
known restrictions, 2-6
mapping user IDs with DB2/VMT, 15-6
B
binary data, non-character, 14-27
Bind Package Stored Procedure
DB2/400, 5-5
DB2/OS390, 5-3
DB2/UDB, 5-7
DB2/VM, 5-9
BIND privilege
DB2/OS390, 5-3
DB2/UDB, 5-7
DB2/VM, 5-9
bind variables, 14-36
restrictions, 2-10
BINDADD authority, 11-9
BINDADD privilege
DB2/OS390, 5-3
DB2/UDB, 5-7
DB2/VM, 5-9
BINDAGENT privilege
DB2/OS390, 5-3
DB2/UDB, 5-7
DB2/VM, 5-9
binding the DRDA package
  backward compatibility, 16-6
  DB2/400, 5-5
  DB2/OS390, 5-3
  DB2/UDB, 5-7
  DB2/VM, 5-9
  on DB2/UDB, 11-9
  on DRDA server, 11-8
  pre-binding checklist, 11-10
books
  other recommended reading, 3-5
Bourne shell
  setting ORACLE_HOME variable, 4-5
bug
  debugging, 13-11, 17-3, 17-8
  debugging with Startup Shell Script, 11-4
  number 205538, known restrictions, 2-9
bugs fixed in this release, 2-4

C

C shell
  setting ORACLE_HOME variable, 4-5
call
  a CICS or IMS transaction, 14-7
  DB2 stored procedure, 14-5, 14-6
  DRDA_DISABLE_CALL, 11-10, B-2
  empproc stored procedure, 14-5
  Oracle Call Interfaces, 17-2
  PL/SQL, 14-6
  stored procedure, 14-3, 14-4
  to stored procedure
  known restrictions, 2-7
capabilities
  DRDA server, 14-24
catalog views, 14-39
CCSID
  65535 as the default for all tables created,  C-5
  codepage mapping facility,  D-9
description, D-4
  DRDA_OVERRIDE_FROM_CODEPAGE, C-8
  external mapping to Oracle character sets
  supported, 2-2
  CD-ROM drive requirements, 3-2
  CEIL SQL function, 14-25, F-1
changed parameters, 16-4
changes in this release
  main topic, 2-3
  bugs fixed, 2-4
gateway parameters, 2-3
  IBM DB2 Version 5.1 EBCDIC and ASCII Tables, 2-3
  IBM DB2/UDB supported, 2-3
  Oracle server dependencies, 2-3
  read-only support, 2-3
  TNSNAMES.ORA, 2-3
  CHAR_TO_ROWID function, F-1
  CHARACTER SET parameter, D-4
description, D-3
  character sets
  and codepage map facility, D-9
  ASCII, 2-3
  CCSID, D-4
codepage, C-5
  EBCDIC, 2-3
  limitation, 2-7
  supported, 2-2

character string
  converting datatypes, 14-28
  performing operations, 14-28
checklist
  configuring the communications interfaces, 6-2,
  8-2
  configuring the DRDA server, 5-2
gateway configuration, 11-2
  gateway installation, 4-3
  installation overview, 4-3
  Oracle Net, 12-2
CHECKSUM command, 1-6, 12-6
CHR SQL function, 14-25, F-1
CICS transaction, 14-7
clauses
  CONNECT TO, 13-2
  GROUP BY, 14-26
  HAVING, 14-26
  ORDER BY, 14-26
  SQL
    DELETE, 14-34
    INSERT, 14-34
    SELECT WHERE, 14-34
UPDATE, 14-34
USING, 13-2
VALUES
  functions not allowed by DB2, 14-34
WHERE, 14-26
  known restrictions, 2-9
WHERE CURRENT OF CURSOR
  known restrictions, 2-9
code tracing, C-17
codepage map facility
  configuring support for character sets, known
    restrictions, 2-7
  for data translation, D-9
  supported by gateway, 2-2
codepage, overriding, C-8
coercion
  of data, 14-24
collection privilege - CREATE IN, 5-3, 5-7, 5-9
column
date columns, TO_DATE function, 14-34
Oracle ROWID
  known restrictions, 2-9
  supported in a result set, 1-16
commands
AGW ADD USERID, 15-6
AGW DELETE USERID, 15-6
CHECKSUM, 1-6, 12-6
COPY
  known restrictions, 2-6
  Oracle9i server to DRDA server, 13-10
COPY (SQL*Plus command), 13-10
CREATE DATABASE LINK
  DB2/VM, 5-9
DESCRIBE
  known restrictions, 2-6
EXECUTE, 1-8
EXPLAIN PLAN, 17-8
INSERT
  known restrictions, 2-6
  INSERT, not supported, 13-10
lanscan, 7-7, 9-7
commit confirm protocol, 1-9
Communication Database (CDB) tables, DDF, 5-4
communications requirements, 3-4
Communications Server
  configuring server profiles, 8-3
    profiles
      creating, 8-3
  compatible SQL set operators and clauses, 14-26
  concatenation restrictions, 2-7
  concurrent connections
    APPC, 3-2
    TCP/IP, 3-2
configuration
  SNA sample file, 6-4
configuring
  additional DRDA servers, 11-14
  AS/400 communications, 5-5
  AVS, 5-9
  binding DRDA package, 11-10
  checklist for gateway, 11-2
  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, 5-5, 5-7
  DB2/OS390, 5-3
  DB2/UDB, 5-7
  host workstation for gateway, 11-4
  MVS VTAM, 5-3
  Oracle integrating server, 11-13
  Oracle Net, 12-4
  other Oracle servers, 11-13
  SNA server
    creating profiles, 8-3
    testing connection, 8-6
  SNAPlus2 on HP-UX, 9-1
  SunLink SNA Peer-to Peer product, 6-3
TCP/IP
  DB2/400, 5-5
  DB2/OS390, 5-3
  DB2/UDB, 5-7
  DB2/VM, 5-9
  under Solaris, 10-2
VM VTAM, 5-9
CONNECT authority, 11-9, 15-6
CONNECT BY not supported
  known restrictions, 2-10
CONNECT TO clause, 13-2
contact information
  in configuring SNAplus2, 7-8, 9-9
conversion errors, C-5
convert
character string, 14-28
datatypes
   DRDA to Oracle datatypes, 14-26
   DATE, 14-29
   floating point to integer, 14-34
   inbound user ID, 15-6
   into most suitable datatype, 14-35
   SQL, 1-14
   to the numeric datatype, 14-34
CONVERT SQL function, 14-25, F-1
converter, protocol, 1-6
COPY command
   Oracle9i server to DRDA server, 13-10
COPY privilege
   DB2/OS390, 5-3
   DB2/UDB, 5-7
   DB2/VM, 5-9
COPY SQL*Plus command, 13-10
substituted for INSERT, known restrictions, 2-6
copying data
   from the DRDA server, 13-10
   from the Oracle server to DRDA server, 13-10
COS SQL function, 14-9, 14-25, F-1
COSH SQL function, 14-25, F-1
COUNT function, 14-35
CP (Control Point for the Node), 7-3, 9-3
CPI-C
   communication error messages, 17-4, 17-7
   create Side Information Profile, 7-14, 9-15
   profiles, creating, 7-12, 9-13
   requires Side Information File, 6-4
   routine, 17-4
   side information profile, 7-14, 9-15
CREATE command
   supported by COPY, 13-10
CREATE DATABASE LINK command, 13-2
   DB2/VM, 5-9
CREATE IN privilege
   DB2/OS390, 5-3
   DB2/UDB, 5-7
   DB2/VM, 5-9
CREATE PUBLIC DATABASE LINK
   privilege, 11-8, 11-13
CREATE TABLE statement, 1-7
CREATETAB authority, 11-9
creating
   a new directory, 4-5
   CPI-C and mode definitions, 7-12, 9-13
database link, 13-2
   link station, 7-7, 9-8
   local LU names, 7-10, 9-11
   partner LUs, 7-11, 9-12
   remote node definition, 7-7, 9-8
   SNA definitions for the gateway, 7-3, 9-3
   SNAplus2 profiles, 7-2, 9-2
cursor
   defining the number of, 14-39
   number of, C-11
   stability, C-7
customizing
   Startup Shell Script, B-3

D
data coercion, 14-24
data control language (DCL), 1-7
DATA datatype, 14-29
data definition language (DDL), 1-7
data dictionary
   improved in this release, 2-2
   support, 11-11
   using, 14-39
views
   ALL_DB_LINKS, 13-3
   considerations for migration from previous
   releases, 16-6
   emulation on DRDA server, 14-38
   for DB2/VM and DB2/UDB not supported, A-2
   list and descriptions, A-4
   supported for DB2/OS390 and DB2/400
   servers, A-2
   USER_DB_LINKS, 13-3
database
   catalogs, 14-38
database link
   behavior, 14-7
binding the gateway package, 11-8
creating, 13-2
defining and controlling, 15-3
dropping links, 13-3
examining, 13-3
guidelines, 13-3
limits, 13-3
processing, 13-2
public, 11-13
suffix, 14-2
to identify the gateway, 1-13
database triggers, 1-5
datatype
code string, 14-27
column (ALL_TAB_COLUMNS), A-11
column (USER_TAB_COLUMNS), A-22
column (USER_TAB_COLUMNS), A-22
conversion
DRDA to Oracle datatypes, 14-26
no control over, 14-34
converting character string, 14-27, 14-28
data and time, 14-29
differences between Oracle server and DRDA
databases, 14-1
DRDA server datatypes list, 14-26
mapping, 14-26
numeric, 14-34, 14-35
operations, numeric, 14-34
Oracle datatypes RAW and LONG RAW, 14-28
restrictions, 14-26
size and value limitations, 14-26
datatypes
DATE, 14-29
GRAPHIC, 14-28
LONG, 14-28
LONG RAW, 14-28
Oracle and IBM DATE, 14-29
Oracle DATE, 14-29
RAW, 14-28, C-5
TIME, 14-29
TIMESTAMP, 14-29
VARCHAR, 14-28
VARGRAPHIC, 14-28
date
date columns, TO_DATE function, 14-34
DELETE
statement, 14-32
HS_NLS_DATE_FORMAT parameter, 14-32, 14-33
INSERT
statement, 14-32
operations, 14-29
SELECT statement, 14-32
TO_DATE function, 14-32
UPDATE
statement, 14-32
date arithmetic
known restrictions, 2-8
DATE datatype, 14-29
DB_DOMAIN parameter
known restrictions, 2-7
DB2
02pcg.sql granting authority, 11-12
alias objects
known restrictions, 2-6
aliases, C-5
CICS, 14-7
configuring DB2/OS390, 5-3
data access, 1-8
Distributed Data Facility (DDF), 5-4
DRDA_DESCRIBE_TABLE compatibility, C-5
IBM DB2 Version 5.1 ASCII Tables, 2-3
IMS, 14-7
native SQL, 1-7
native stored procedures, 1-8
procedural feature considerations, 14-7
SPUFI utility, 5-4
SQL statements, 14-36
statements
CREATE TABLE, 1-7
stored procedures, 1-7
DB2/400
catalog view, 14-39
configuring, 5-5, 5-7
data dictionary views supported by
gateway, A-2
DRDA_DEFAULT_CCSID, C-5
DRDA_DISABLE_CALL, C-6
DRDA_ISOLATION_LEVEL, C-7
DRDA_OPTIMIZE_QUERY, C-8
DRDA_PACKAGE_COLLID, C-9
user ID mapping, 15-7
DB2IND sample DB2 stored procedure, G-2
DB2/OS390
  catalog view, 14-39
  configuring, 5-3
  data dictionary views supported by
gateway, A-2
  DRDA_DISABLE_CALL, C-6
  DRDA_ISOLATION_LEVEL, C-7
  DRDA_OPTIMIZE_QUERY, C-8
  user ID mapping, 15-5
V6 and V7 stored procedures supported, 2-2
DB2/UDB
  binding packages, 11-9
  binding the gateway package, 11-10
  catalog view, 14-39
  configuring, 5-7
  data dictionary views not supported, A-2
  DRDA_ISOLATION_LEVEL, C-7
  DRDA_OPTIMIZE_QUERY, C-8
  incompatibilities with gateway, 2-5
  known restrictions, 2-8
  supported, 2-3
  user ID mapping, 15-7
DB2/VM
  catalog view, 14-39
  configuring, 5-9
  data dictionary views not supported, A-2
  database and SQL functions, 14-20
  DRDA_ISOLATION_LEVEL, C-7
  DRDA_OPTIMIZE_QUERY, C-8
  DRDA_PACKAGE_OWNER, C-11
  instance, DRDA location name, 5-10
  server machine, 15-6
  user ID mapping, 15-6
DBCS (double-byte character set)
  known restrictions, 2-7
DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE
  function, 14-36
DD basic tables, known restrictions, 2-6
DDF
  DB2 (Distributed Data Facility), 5-4
  subsystem, 5-3
DDL
  statement, 14-36
debug
  gateway, 17-8
  library, 17-8
debugging
  error codes, 17-3
  SQL tracing, 17-8
  with Startup Shell Script, 11-4
  your application, 13-11
DESCRIBE
  character string operations, 14-28
  command, known restrictions, 2-6
diagnostic parameter, C-16
dictionary
  mapping, 1-6
  see also data dictionary
  tables, 14-38
DICTIONARY view, A-14
disk space requirements, 3-3
distributed
  operations, DB2, 5-4
  queries
    example of, 13-7
    two-phase commit, 13-8
  transaction, C-12
distributed applications
  support for, 1-19
distributed data facility (DDF), 5-4
distributed database, 12-3
distributed DRDA transactions, 13-8
distributed processing, 12-3
distributed query optimizer (DQO), 13-6
  DRDA-specific parameters, C-8
dmesg, 7-7
documentation
  recommended, 3-5
double-byte character set (DBCS)
  known restrictions, 2-7
DQO
also see distributed query optimizer, 13-6
DRDA-specific parameters, C-8
drc error code, 17-3
DRDA
catalog, 14-39
database requirements, 3-3
defining number of cursors, 14-39
gateway package considerations, 11-10
location name
for DB2/UBD instance, 5-8
for DB2/VM instance, 5-10
mode
IBM RDB, 7-13, 9-14
session security options, 15-5
DRDA Application Server Function, 1-17
DRDA server
accessing, 11-14
and dependent LUs, 7-3, 9-3
and independent LUs, 7-3, 9-3
and Side Information Profiles, 7-2, 9-2
architecture, 1-11
capabilities, 14-24
character sets
known restrictions, 2-7
client applications initiate conversations using
dependent LUs, 7-3, 9-3
configuring
DB2/400, 5-5
DB2/OS390, 5-3
DB2/UBD, 5-7
DB2/VM, 5-9
considerations for binding packages, 16-6
copying data
from Oracle server, 13-10
to Oracle server, 13-10
database link behavior, 14-8
default TP name, 7-15, 9-16
functions, 14-24
languages and character sets in
configuration, D-9
Logical Unit (LU), 6-6, 7-17, 8-7, 9-18
partner LU, 7-11, 9-12
stored procedures, 14-5
VTAM definitions and local LU names, 7-10,
9-11
DRDA_CAPABILITY parameter, 14-24
DRDA_CMSRC_CM_IMMEDIATE parameter, C-4
DRDA_CODEPAGE_MAP parameter, D-9
DRDA_COMM_BUFSIZE parameter, C-3
DRDA_CONNECT_PARM (SNA format)
parameter, C-4
DRDA_CONNECT_PARM (TCP/IP format)
parameter, C-4
DRDA_CONNECT_PARM parameter, 17-4
determine SNA V9 value for worksheet, 6-6
DRDA_DEFAULT_CCSID parameter, C-5
DRDA_DESCRIBE_TABLE parameter, C-5
known restrictions, 2-6
DRDA_DESCRIBE_TABLE=FALSE initialization
parameter
known restrictions, 2-6
DRDA_DISABLE_CALL parameter, 11-10, C-6
DRDA_FLUSH_CACHE parameter, C-6
DRDA_ISOLATION_LEVEL parameter, C-7
DRDA_LOCAL_NODE_NAME parameter, C-7
DRDA_OPTIMIZE_QUERY parameter, 13-6, C-8
DRDA_PACKAGE_COLLID parameter, 17-4, C-9
DRDA_PACKAGECONSTOKEN
parameter, C-10
DRDA_PACKAGE_NAME parameter, 11-10, 17-4,
C-10
DRDA_PACKAGE_OWNER parameter, C-11
DRDA_PACKAGE_SECTIONS parameter, 14-39,
C-11
DRDA_READ_ONLY parameter, 13-9, C-12
DRDA_RECOVER_USERID, 5-8
DB2/400, 5-5
DB2/OS390, 5-4
DB2/VM, 5-9
DRDA_RECOVERY_PASSWORD parameter, 5-8,
C-12
DB2/400, 5-5
DB2/OS390, 5-4
DB2/VM, 5-9
DRDA_RECOVERY_USERID parameter, 5-8, C-12
DB2/400, 5-5
DB2/OS390, 5-4
DB2/VM, 5-9
DRDA_REMOTE_DB_NAME parameter, C-13
DRDA_SECURITY_TYPE parameter, C-13
DROP DATABASE LINK statement, 13-3
dropold.sql script, 11-11
DSPRDBDIRE command, 5-6
DUMP SQL function, 14-25, F-1
dynamic dictionary mapping, 1-6

E

EBCDIC
  character set support, D-7
code page, D-5
DRDA server CCSID, D-12
sort order, 14-25
tables, known restrictions, 2-3
translated to ASCII, 14-27
EMP
  system-wide synonym, 13-5
table, 13-10
empproc
  stored procedure, 14-5
encryption types for Advanced Security, 12-7
enhancements in this release, 2-2
environment
  heterogeneous, 13-9
environmental variable
  NLS_LANG, D-3
  ORA_NLS33, D-3
  ORACLE_HOME, 4-5
environmental variables
  $DISPLAY, 7-4, 9-4
err
  DRDA_OVERRIDE_FROM_CODEPAGE usage
    note, C-8
    mapped error example, 17-5
errmc
  CPI-C routine, 17-4
  DRDA_OVERRIDE_FROM_CODEPAGE usage
    note, C-8
  errmc field lists any error tokens, 17-3
error tokens, 17-4
  mapped error example, 17-5
errp
  errp field indicates program that detected
    error, 17-3
error
  12660 (test error for Advanced Security), 12-7
  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, 17-4
condition, 17-2
conversion, C-5
date, D-7
detected
  by Oracle integrating server, 17-2
  by server database, 17-4
  by the gateway, 17-3
  in DRDA software, 17-3
  in the DRDA software, 17-3
drc= field
  -300xx, 17-7
  -7xx, 17-6
HGO-00706, 17-3
host database, 17-7
interpreting error messages, 17-2
mapped sqlstate, 17-5
messages, 6-8, 7-19, 8-9, 9-20, 14-28
messages & codes, 17-1
number, 17-2
obsolete parameters, 16-5
ORA-00001, index constraint violated, 17-5
ORA-00942
  mapped error example, 17-5
  object does not exist, 17-5
  object name too long, 17-5
ORA-01017, logon denied, 17-5
ORA-01031, insufficient privileges, 17-5
ORA-01460, invalid CCSID, 17-5
ORA-01476, divide by zero, 17-5
ORA-02019, 17-2
ORA-2025, when using INSERT
  command, 13-10
ORA-28500 (was ORA-09100), 17-3, 17-4
ORA-28501
  communication error, 17-4
  was ORA-09101, 17-4
ORA-28527
DRDA_OVERRIDE_FROM_CODEPAGE usage note, C-8
ORA-9100 to ORA-9199, 17-2
Oracle mapped error codes, 17-5
specific gateway error codes, 17-6
tokens, 17-3, 17-4
translation, 14-28
while binding the gateway package, 11-8
with Native Semantics, 14-24
error array (errd), See errd
errp
DRDA_OVERRIDE_FROM_CODEPAGE usage note, C-8
mapped error example, 17-5
EXCEPT set operator, 14-26
execute authority, 15-6
EXECUTE command, 1-8
EXECUTE privilege
DB2/OS390, 5-3
DB2/UDB, 5-7
DB2/VM, 5-9
exits
gateway local date, 14-33
EXP SQL function, 14-25, F-1
EXPLAIN PLAN command, 17-8
EXPLAIN_PLAN table, 1-16
export encryption algorithms, 12-6

F
FDS_CLASS parameter, C-14
FDS_CLASS_VERSION parameter, C-14
FDS_INSTANCE parameter, C-15
features
Oracle Snapshots, 13-9
features of the gateway
application development and end-user tools, 1-19
application portability, 1-17
columns supported in a result set, 1-16
distributed applications supported, 1-19
EXPLAIN_PLAN improvement, 1-16
fetch reblocking, 1-15
heterogeneous database integration, 1-16
heterogeneous services architecture, 1-15
large base of data access, 1-17
main topic, 1-15
minimum impact on existing systems, 1-17
Native Semantics, 1-16
Oracle9i passthrough supported, 1-15
performance enhancements, 1-15
remote data access, 1-18
retrieving result sets through passthrough, 1-16
support for TCP/IP, 1-16
fetch array size, with
HS_FDS_FETCH_ROWS, C-15
fetch reblocking, 1-15, 14-3
fetched date, C-16
fields
errmc, lists any error tokens, 17-3
errp, indicates program that detected error, 17-3
file member name, 13-5
file members
accessing AS/400 files, 13-5
files
initsid.ora
communication errors, 17-4
gateway error -700, 17-6
listener.ora, 11-5, 12-5
sample, B-1
listener.ora, B-4
tnspames.ora, B-3
sna_omnia.cfg, 7-3, 9-3
sna_node.cfg, 7-3, 9-3
tnspames.ora, 11-13, 11-14
modifying, 12-5
VSAM, 14-7
FLOOR SQL function, 14-25, F-1
FOR BIT DATA
DRDA_DEFAULT_CCSID, C-5
option, 14-28
free session,
DRDA_CMSRC_CM_IMMEDIATE, C-4
functions
ABS, F-1
ACOS, F-1
ADD_MONTHS, F-1
ASCII, F-1
ASIN, F-1
ATAN, F-1
ATAN2,  F-1
CEIL,  F-1
CHAR_TO_ROWID,  F-1
CHR,  F-1
CONVERT,  F-1
COS,  14-9,  F-1
COSH,  F-1
COUNT,  14-35
DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE,  14-36
DECODE,  F-1
DRDA server,  14-24
DUMP,  F-1
EXP,  F-1
FLOOR,  F-1
GREATEST,  F-1
HEXTORAW,  F-1
INITCAP,  F-2
INSTR,  F-2
INSTRB,  F-2
LAST_DAY,  F-2
LEAST,  F-2
LENGTH,  F-2
LENGTHB,  F-2
LN,  F-2
LOG,  F-2
LOWER,  F-2
LPAD,  F-2
LTRIM,  F-2
MOD,  F-1
MONTHS_BETWEEN,  F-1
NEW_TIME,  F-1
NEXT_DAY,  F-1
NLS_INITCAP,  F-1
NLS_LOWER,  F-1
NLS_UPPER,  F-1
NLSSORT,  F-1
POWER,  F-1
RAWTOHEX,  F-1
REPLACE,  F-1
ROUND,  F-1
ROWIDTOCHAR,  F-1
RPAD,  F-1
RTRIM,  F-1
SIGN,  F-1
SIN,  F-1
SOUNDEX,  F-1
SQL
   SUBSTR,  14-24
   TANH,  F-2
SQRT,  F-2
STDDEV,  F-2
SUBSTR,  F-2
   known restrictions,  2-6
SUBSTRB,  F-2
SYSDATE,  F-2
TAN,  F-2
TO_CHAR,  F-2
TO_DATE,  14-30,  14-32,  14-33,  F-2
TO_DATE,  main topic,  14-34
TO_LABEL,  F-2
TO_MULTI_BYTE,  F-2
TO_NUMBER,  F-2
TO_SINGLE_BYTE,  F-2
TRANSLATE,  F-2
TRUNC,  F-2
UID,  F-2
UPPER,  F-2
USER,  F-2
USERENV,  F-2
VARIANCE,  F-2
VSIZE,  F-2

G

G4ddtab.sql script,  11-11
G4ddview.sql script,  11-11
G4drdrv
   former startup program,  11-5
gateway
   accessing
      from other Oracle servers,  11-13
      main topic,  13-4
advantages
   main topic,  1-3
   migration and coexistence,  1-9
   multisite transactions,  1-9
   security,  1-10
   server technology and tools,  1-8
   site autonomy,  1-9

Index-11
two-phase commit, 1-9
and Oracle tools, 1-14
and stored procedures (Oracle and non-Oracle), 1-8
application tools, 1-19
architecture, 1-11
authenticating logons, 15-2
benefits of integration with 9i server, 1-6
binding DRDA packages, 11-8
components, 1-12
configuration and the Startup Shell Script, 11-4
definition of terms, 1-10
de-installing, 4-8
DRDA package considerations, 11-10
error codes, 17-6
errors detected, 17-3
features, 1-15
main topic, 1-15
how to access, 13-4
installation
log file, 4-7
steps, 4-4
steps via Oracle Universal Installer, 4-7
interface, 1-13
local date exit, 14-33
logging, C-16
migration problems, 16-3
name, setting up, with Sunlink, 6-3
parameter, 5-10
performance, 14-24
performance enhancements, 1-15
performing distributed queries, 13-6
read-only option, 13-9
service name entries in the tnsnames.ora, 16-6
settings for security options, 6-7, 7-17, 8-8, 9-19
SQL differences, 1-14
startup time, 11-4
supported languages, D-9
tracing, C-16
tracing SQL statements, 13-11
using, 13-1
with other Oracle products
SQL*Plus, 1-9
Gateway Initialization File
backup and recovery of gateway configuration, 11-12
configuring for binding the DRDA gateway package, 11-8
defining userids and passwords, 15-8
encrypting passwords, 15-9
if errors are reported, 11-8
modifying, C-2
parameters
changes in release 9.0.1, 2-3
DRDA_READ_ONLY, 13-9
LOG_DESTINATION, 17-9
LOG_DESTINATION parameter, tracing, 17-9
new since V4 gateway, 16-4
ORACLE_DRDA_TCTL, 17-9
stored in init.ora, C-2
TRACE_LEVEL, 17-9
parameters, list, C-3
sample, B-2
see also parameters
Gateway System Identifier (SID), 11-4
GCS virtual machine, 15-6
GLOBAL_NAMES
known restrictions, 2-7
GRANT, 15-6
GRANT statement, 11-13
granting authority to a package for DB2, 11-12
GRAPHIC datatype, 14-28
graphic string operations
unsupported, 14-28
grc error code, 17-3
GREATEST SQL function, 14-25, F-1
GROUP BY clause, 14-26
GROUPBY SQL function, 14-25
GTWS_BIND_PKG
internal stored procedure, 11-10
stored procedure, 11-8

H

hardware requirements, 3-2
HAVING clause, 14-26
HAVING SQL function, 14-25
heterogeneous database integration, 1-16
Heterogeneous Services (HS), see HS, 1-3
HEXTORAW SQL function, 14-25, F-1
HGO-00706 error, 17-3
host
  DRDA server, SNA and TCP/IP protocols, 1-11
  networking needs, 3-4
  performing character string operations
  on, 14-27
  relationship to gateway and Oracle Server, 1-11
  security validation required to begin database
  link, 6-6, 7-17, 8-7, 9-18
  SNAP-IX definitions, 7-4
host database error, 17-7
host Logical Unit (LU), 6-6, 7-17, 8-7, 9-18
host variable, 14-26, 14-28
HP9000 host, 9-4, 9-11
  invoking xsnapadmin, 9-4
HS (Heterogeneous Services)
  architecture features, 1-15
  Oracle Net considerations, 16-6
HS= (TNSNAMES parameter for Oracle Net), 12-5
  gateway migration problems, 16-3
HS_DB_DOMAIN parameter
  known restrictions, 2-7
HS_FDS_FETCH_ROWS parameter, C-15
HS_NLS_DATE_FORMAT parameter, D-7
  14-33
HS_NLS_DATE_LANGUAGE, D-7
HS_RPC_FETCH_REBLOCKING parameter, 1-15, 14-3
HS_RPC_FETCH_SIZE parameter, 1-15, 14-3
IBM Communications Server for AIX
  and SNA security option, 6-8, 7-18, 8-9, 9-20
  tailoring Startup Shell Script for Version
  9.0, 11-6
  Version 5 required patches, 3-4
IBM RDB, 7-15, 9-16
  DRDA mode, 7-13, 9-14
  implementation, 1-12
  implicit data conversion, 14-24
  implicit protocol conversion, 1-6
  IMS transaction, 14-7
  inbound connections
  processing, 15-4
  init parameter, in Startup Shell Script, 11-5
INITCAP SQL function, 14-25, F-2
initdrdahoa1.ora
  sample, B-2
  see also Gateway Initialization File parameters
initdrdahoa1.sh
  Also See Startup Shell Script, 11-4
  sample, B-3
initialization parameters
  new since V4 gateway, 16-4
  see also Gateway Initialization Files, parameters
  where stored, 11-4
initsid.gtwboot, 16-2
  parameters formerly stored in the gateway
  instance directories, 11-4
initsid.ora file
  communication errors, 17-4
  gateway error -700, 17-6
  migrating, 16-2
  NLS parameters, D-6
  parameters formerly stored in the gateway
  instance directories, 11-4
input bind variables, 14-30
INSERT
  known restrictions, 2-6
  operation, 14-2
  Oracle SQL command, known restrictions, 2-6
  read-only gateway, 13-9
  SQL clause, 14-34
  statement, 14-32, 14-36
INSERT command
  known restrictions, 2-6
  supported by COPY, 13-10
INSERT command, not supported, 13-10
installation
  checklists, 4-3, 5-2, 6-2, 8-2, 11-2, 12-2
  overview, 4-4
  installing the gateway
  from CD, 4-4
INSTR SQL function, 14-25, F-2
INSTRB SQL function, 14-25, F-2
internal tracing, C-17
Internet support, 1-6
INTERSECT

Index-13
SQL set operators and clauses, 14-26
IPC
adapter, B-4
ISO standard, 1-7
isolation level, C-7

J
JOIN
capability, 1-5
JOIN SQL statement, 14-2

K
keywords
LISTENER, B-4
known problems, 2-5
compatibility with DB2/UDB, 2-5
known restrictions, 2-6
accessing DB2 alias objects, 2-6
AVS mapping user IDs, 2-6
bind variables become SQL parameter markers, 2-10
binding the DRDA gateway package on DB2/UDB, 2-8
character set limitation, 2-7
CONNECT BY not supported, 2-10
datatype limitations, 2-7
date arithmetic, 2-8
DD basic tables and views, 2-6
dictionary views not provided for DB2/VM or DB2/UDB, 2-9
DRDA server character sets, 2-7
GLOBAL_NAMES parameter, 2-7
INSERT (Oracle SQL command), 2-6
LONG datatype in SQL*Plus, 2-8
null values and stored procedures, 2-7
Oracle ROWID column, 2-9
row length limitation, 2-8
SAVEPOINT, 2-7
single gateway instances per DRDA network interface, 2-9
SQL*Plus DESCRIBE command, 2-6
string concatenation, 2-7
SUBSTR function post-processed, 2-6
Korn shell
setting ORACLE_HOME variable, 4-5

L
LANGUAGE parameter, D-6
see also Gateway Initialization File
languages, 1-8
SQL*Plus, 1-9
lanscan command, 7-7, 9-7
LAST_DAY function, F-2
LEAST SQL function, 14-25, F-2
LENGTH SQL function, 14-25, F-2
LENGTHB SQL function, 14-25, F-2
link station
creating, 7-7, 9-8
parameters, 7-9, 9-10
testing SNA connection, 7-19, 9-21
link, also see Database Link, 14-7
linkage conventions
    SIMPLE WITH NULLS, 14-7
linkname, 11-13
listener, B-4
LISTENER keyword, B-4
listener.ora file, 11-5, 12-5
    sample, B-4
literal
    character literals, 14-29
date, 14-29
    specific datatype, 14-26
    TO_DATE, format support, 14-34
LN SQL function, 14-25, F-2
log file, gateway installation, 4-7
LOG SQL function, 14-25, F-2
LOG_DESTINATION parameter, 17-9, C-16, C-17
logging, C-16
Logical Unit (LU), 6-6, 7-17, 8-7, 9-18
Login as DBA, 4-4
LONG datatype, 14-28
LONG RAW datatype, 14-28
LOWER SQL function, 14-25, F-2
LPAD SQL function, 14-25, F-2
LTRIM SQL function, 14-25, F-2
LU
dependent, 7-3, 9-3
**additional SNAPlus2 configuration needed**, 7-3, 9-3

**Side Information Profiles**, 7-3, 9-3

**independent vs. dependent**, 7-3, 9-3

**local LU alias**, 7-10

**CPI-C destination**, 7-15, 9-16

**Ethernet parameters**, 9-11

**local names**, 9-11

**creating**, 7-10, 9-11

**names**, 7-15, 9-16

**partner creating**, 7-11, 9-12

**SNAP-IX profiles**, 7-2

**SNAPPlus2 profiles**, 9-2

**traffic type, in configuring SNAplus2**, 7-8, 9-9

**LU name**, 15-5

**LU6.2**

**local LU profile**, 8-4

**profile types**, 8-3

**side information profile**, 8-5

**SNA APPC**, 6-1, 7-1, 8-1, 9-1

**M**

**MAC address**

- in SNA configuration, 7-9
- in SNAplus2 configuration, 9-10

**mapped sqlstate errors**, 17-5

**mapping user IDs**

- AVS, 15-6
- known restrictions, 2-6
- migrating the gateway instance, 16-6
- migration problems, 16-3

**MINUS**

- set operator, 14-26
- SQL set operators and clauses, 14-26

**MOD SQL function**, 14-25, F-1

**mode**

- name, 7-15, 9-16
- profiles, creating, 7-12, 9-13

**Mode Profile**

- SNA Server profile, 8-4

**Mode_name**

- description, 6-5

**MONTHS_BETWEEN function**, F-1

**MVS VTAM**

- configuring DB2/OS390, 5-3

**N**

**National Language Support**

- initsid.ora parameters, D-6
- overriding codepage, C-8
- overview, D-1

**Native Semantics**

- gateway architecture, 1-16
- parameters, 14-26
- with SUBSTR function, known restrictions, 2-6

**network**

- attachment, 3-2
- Oracle Net configuration, 12-4
- requirements, 3-2
- transparency, 1-4

**NEW_TIME function**, F-1

**NEXT_DAY function**, F-1

**NLS**

- Also See National Language Support, D-1
- DRDA server character sets, D-9

**NLS parameters**

- configuration on client and Oracle servers, D-4
- NLS_INITCAP SQL function, F-1
- NLS_LANG
  - environmental variable, D-3
  - parameter default setting, D-8
  - syntax, D-8
  - server-side parameter, D-3
- NLS_LOWER SQL function, F-1
- NLS_UPPER function, F-1
- NLS_UPPER SQL function, F-1
- NLSSORT SQL function, 14-25, F-1

**non-character binary data**, 14-27

**null**

- rows, 14-35
- values, 14-35

**number of cursors**, C-11

**numbers**

- concatenation restrictions, 2-7
- numeric datatype, 14-34, 14-35

Index-15
Index-17

processing, 12-3
editing to set up security test, 12-7
Heterogeneous Services (HS) facility, 12-3
introduction, 12-3
operating system authentication, 15-3
overview, 12-3
product version, 2-2
purpose, 1-12
requirements, 3-4
sample files, B-1
sample listener.ora file, B-4
support for CHECKSUM and encryption, 12-6
terminology, 12-4
client, 12-4
driver, 12-4
host, 12-4
network, 12-4
protocol, 12-4
TNS connect descriptor specification, 13-2

Oracle ROWID column
known restrictions, 2-9
Oracle server dependencies, 2-3
Oracle Universal Installer
de-installing the gateway, 4-8
starting, 4-6
steps to install the gateway, 4-7
upgrading for installation, 4-6
ORACLE_DRDA_TCTL parameter, 17-9, C-16
ORACLE_DRDA_TRACE parameter, C-17
ORACLE_HOME
environmental variable, 4-5
for the gateway, 4-5
ORACLE2PC table, 11-9, 11-10, 11-12, 13-8, C-11
DB2/400, 5-5
DB2/OS390, 5-3
DB2/UDB, 5-7
DB2/VM, 5-9
Oracle9i Server
relationship to host, 1-11
services, 1-5
database triggers, 1-5
distributed capabilities, 1-5
distributed query optimization, 1-5
SQL, 1-5
stored procedures, 1-5
two-phase commit protection, 1-5
triggers, 13-9
Oracle9i server
copying data
from DRDA server, 13-10
to DRDA server, 13-10
Oracle9i database server, description, 1-3
ORADRDA.ORACLE2PC table, 13-8
ORAIND sample DB2 stored procedure, G-4
oraproc1, 14-4
stored procedure, 14-3
oraproc2
stored procedure, 14-3
ORARECOV user ID, 5-8, C-12
DB2/400, 5-5
DB2/OS390, 5-4
DB2/VM, 5-9
ORDER BY clause, 14-26
ORDERBY SQL function, 14-25
OS/390 (MVS)
configuring DB2/OS390, 5-3
overriding codepage, C-8

P

package
collection id, C-9
consistency token, C-10
privileges - BIND, COPY, and EXECUTE, 5-3, 5-7, 5-9
packed decimal, 14-35
parameter
changed, 16-4
checking settings, 11-10
diagnostic, C-16
gateway, 5-10
init, in Startup Shell Script, 11-5
link station, 7-9, 9-10
Native Semantics, 14-26
new since V4 gateway, 16-4
obsolete since V4 gateway, 16-5
renamed since V4 gateway, 16-5
setting up trace parameters, 17-8
Startup Shell Script, new since V4 gateway, 16-3
parameter file
tailoring to configure the host, 11-4
parameters
CHARACTER SET, D-4
DB_DOMAIN
    known restrictions, 2-7
DRDA_CAPABILITY, 14-24
DRDA_CODEPAGE_MAP, C-3, C-14, D-9
DRDA_DESCRIBE_TABLE
    known restrictions, 2-6
DRDA_DISABLE_CALL, 11-10
DRDA_PACKAGE_NAME, 11-10
DRDA_PACKAGE_SECTIONS, 14-39
DRDA_READ_ONLY, 13-9
DRDA_RECOVERY_PASSWORD, 5-8
    DB2/400, 5-5
    DB2/OS390, 5-4
DRDA_RECOVERY_USERID, 5-8
    DB2/400, 5-5
    DB2/OS390, 5-4
    DB2/VM, 5-9
FDS_CLASS, C-14
FDS_CLASS_VERSION, C-14
FDS_INSTANCE, C-15
Gateway Initialization File
    DRDA_CAPABILITY, C-3
    DRDA_CMSRC_CM_IMMEDIATE, C-4
    DRDA_CODEPAGE_MAP, C-3
    DRDA_COMM_BUFLEN, C-3
    DRDA_CONNECT_PARM, 17-4
    DRDA_CONNECT_PARM (SNA format), C-4
    DRDA_CONNECT_PARM (TCP/IP format), C-4
    DRDA_DEFAULT_CCSID, C-5
    DRDA_DESCRIBE_TABLE, C-5
    DRDA_DISABLE_CALL, C-6
    DRDA_FLUSH_CACHE, C-6
    DRDA_ISOLATION_LEVEL, C-7
    DRDA_LOCAL_NODE_NAME, C-7
    DRDA_OPTIMIZE_QUERY, 13-6, C-8
    DRDA_PACKAGE_COLLID, 17-4, C-9
    DRDA_PACKAGE_CONSTOKEN, C-10
    DRDA_PACKAGE_NAME, 17-4, C-10
    DRDA_PACKAGE_OWNER, C-11
    DRDA_PACKAGE_SECTIONS, C-11
    DRDA_READ_ONLY, C-12
    DRDA_RECOVERY_PASSWORD, C-12
    DRDA_RECOVERY_USERID, C-12
    DRDA_REMOTE_DB_NAME, C-13
    DRDA_SECURITY_TYPE, C-13
    HS_FDS_FETCH_ROWS, C-15
    LOG_DESTINATION, C-16
    ORA_MAX_DATE, C-16
    ORACLE_DRDA_TCTL, C-16
    ORACLE_DRDA_TRACE, C-17
    TRACE_LEVEL, C-17
    HS_DB_DOMAIN
        known restrictions, 2-7
    HS_NLS_DATE_FORMAT, 14-32, 14-33
    HS_RPC_FETCH_REBLOCKING, 1-15, 14-3
    HS_RPC_FETCH_SIZE, 1-15, 14-3
    LOG_DESTINATION, C-17
    LOG_DESTINATION, 17-9
    NLS_LANG, D-8
    OPEN_LINKS, 13-3
    ORA_NLS33, D-8
    PROGRAM, 11-5
    USING, modifying trsnsames.ora file, 12-5
partner LU
    creating, 7-11, 9-12
    creating, also see LU, 7-11
    see LU
partner LU alias, 8-5
partner LU profile, 6-5
Partner_LU_name
description, 6-5
passthrough, 1-7, 1-15, 14-36, 14-38
result sets, 14-36, 14-38
SQL feature, 14-36
patches
    required for IBM Communications Server for AIX, 3-4
performance, 14-24
performance enhancements
    with fetch reblocking, 14-3
PL/SQL
call, 14-6
DRDA stored procedures, 14-5
records, 14-7
routine, 1-8
standard Oracle, 1-8
stored procedure, 14-3
port, 7-7, 9-8
  adding, for SNA, 7-6, 9-6
  name, adding, for SNA, 7-6, 9-7
port number
  Primary, 5-3
    DB2/400, 5-5
    DB2/UDB, 5-7
  Recovery, 5-3
    DB2/400, 5-5
    DB2/UDB, 5-7
post-processed SQL functions
  overview, 14-9
post-processing, 14-24, 17-8
POWER SQL function, 14-25, F-1
PREPARE TRANSACTION statement, 13-8
Primary port number, 5-3
  DB2/400, 5-5
  DB2/UDB, 5-7
privileges
  BIND
    DB2/OS390, 5-3
    DB2/UDB, 5-7
    DB2/VM, 5-9
  BINDADD
    DB2/OS390, 5-3
    DB2/UDB, 5-7
    DB2/VM, 5-9
  BINDAGENT
    DB2/OS390, 5-3
    DB2/UDB, 5-7
    DB2/VM, 5-9
  COPY
    DB2/OS390, 5-3
    DB2/UDB, 5-7
    DB2/VM, 5-9
  CREATE IN
    DB2/OS390, 5-3
    DB2/UDB, 5-7
    DB2/VM, 5-9
  CREATE PUBLIC DATABASE LINK, 11-8, 11-13
data dictionary emulation, 14-38
EXECUTE
  DB2/OS390, 5-3
  DB2/UDB, 5-7
  DB2/VM, 5-9
procedure
  stored, 13-9
    using DRDA server, 14-5
procedures
  DB2 DB2IND, sample, G-2
  DB2 ORAIND, sample for, G-4
processing time, with GROUPBY, HAVING, WHERE, 14-25
processor requirements, 3-2
product installation directory, 4-5
profile
  creating
    CPI-C and Mode profiles, 7-12, 9-13
  name
    see Side Information Profile
  side information, 7-14, 9-15
type
  see Side Information Profile
profile set, 11-14
profile types, 8-3
  Mode Profile, 8-4
PROGRAM parameter, 11-5
protocol
  APPC/LU6.2, 6-1, 7-1, 8-1, 9-1
  commit confirm, 1-9
  communications protocols, 12-3
  converter, 1-6
  definition, 12-4
  implicit protocol conversion, 1-6
  network, 13-5
  Oracle Net Protocol Adapters, 12-3
  PROTOCOL=IPC, sample, B-3
  protocol-independent encryption, 1-6
  two-phase commit, 13-8
protocols
  APPC/LU6.2, 1-4
  IPC, 12-5
  SNA, 1-11
  SPX/IPX, 1-4
  TCP/IP, 1-4, 1-6, 1-11, 1-16
public database link, 11-13
queries, distributed, 13-6

RAW datatype, 14-28, C-5
RAWTOHEX SQL function, 14-25, F-1
read-only gateway option, 13-9, C-12
read-only support, 2-3
rebind, C-6, C-7, C-9, C-11
recommended documentation, 3-5
Recovery port number, 5-3
  DB2/400, 5-5
  DB2/UDB, 5-7
recovery user ID and password, 5-4, 5-8
  DB2/400, 5-5
  DB2/VM, 5-9
remote
  computer, 12-3
  connections, 13-3
  data, 1-5
  data access, 1-18
  database, 5-8, 11-10, 13-2, 13-10, 17-2, 17-7, C-11
    DB2/400, 5-5
    DB2/OS390, 5-4
    DB2/VM, 5-9
database, defining a path, 13-2
database, privileges of user / ID, E-5
DB2 system, 2-6
DRDA database, C-7
DRDA server object information, 2-6
instance, and Oracle stored procedures, 14-3
LU, 8-5
objects, 15-3
Oracle instance, 14-4, 14-5
Oracle servers, 12-3
procedure, 1-8
table, 1-6
TPN, 8-6
transaction program, 3-2, 8-3
userid and password, 13-2
Remote Node definition
  creating, 7-7, 9-8
renamed parameters, 16-5
REPLACE command
  supported by COPY, 13-10
REPLACE SQL function, 14-25, F-1
replication, 13-9
requirements
  hardware, 3-2
  CD-ROM drive, 3-2
  software, 3-3
  restrictions, 2-6
RESULT, 14-6
result sets, 1-16
  columns in, 1-16
return code, 17-2
REVISE_SALARY
  stored procedure, 14-6
ROUND SQL function, 14-25, F-1
ROWID
  Oracle column
    known restrictions, 2-9
ROWIDTOCHAR function, F-1
RPAD SQL function, 14-25, F-1
RTRIM SQL function, 14-25, F-1

sample
  files, B-1
  listener.ora file, B-4
  SQL scripts, 11-11
  tnsnames.ora file, B-3
samples
  Side Information File, SunLink SNA
    Peer-to-Peer, 6-4, 6-6
  SNA configuration file, 6-4
SAP
  dialog box, 7-6, 9-7
  number, 7-9, 9-10
SAVEPOINT
  known restrictions, 2-7
scripts
  dropold.sql, 11-11
  g4ddtab, 11-11
  g4dddview.sql, 11-11
SDL Coaxial network attachment, 3-2
security
Advanced Security, 1-6
DRDA_SECURITY_TYPE, C-13
encryption, 12-6
for session use, 7-15, 9-16
overview, 15-2
site autonomy, 1-9
validation
   for SNA, 6-6, 7-17, 8-7, 9-18
   for TCP/IP, 15-4
SELECT and array size, 1-15
SELECT operation, 14-2
SELECT statement, 14-3, 14-9, 14-38
   read-only gateway, 13-9
SELECT WHERE
   SQL clause, 14-34
semantics, 14-24
server database error, 17-4
server profiles, 6-1, 7-1, 8-1, 9-1
service port number,
   DRDA_CONNECT_PARM, C-4
Service TP
"07F6C4C2", 7-15, 9-16
session
   authorization, 7-15, 9-16
   session, connection, 14-7
set operators
   compatibility, 14-26
   EXCEPT, 14-26
   INTERSECT, 14-26
   MINUS, 14-26
SID
   also see Gateway System Identifier, 11-4
   choosing for gateway, 11-4
   configuring additional DRDA server
      instances, 11-14
   definition, 11-4
   naming suggestion, 11-5
Side Information File
   information needed, 6-4
   sample files, 6-4
   Version 9 sample, 6-6
Side Information Profile, 7-14, 8-5, 9-15, 17-4
   defining, to use dependent LUs, 7-3, 9-3
   description and role, 7-2, 9-2
   specifies Local LU Profile name, 8-4
   side information profile, 8-5
SIGN SQL function, 14-25, F-1
SIMPLE linkage convention, 14-7, G-2
SIN SQL function, 14-25, F-1
single-threaded conversations for dependent
   LUs, 7-3, 9-3
SINH SQL function, 14-25, F-1
site autonomy, 1-9
SMIT (System Management Interface Tool)
   AIX System Management Interface Tool, 8-3
SNA
   APPC, 6-1, 8-1, 9-1
   configuring DB2/400, 5-5
   configuring DB2/OS390, 5-3
   configuring DB2/UDB, 5-7
   configuring DB2/VM, 5-9
   configuring server profiles, 8-3
   connectivity via APPC, 7-1
   conversation security, 6-7, 7-17, 8-8, 9-19
   CPI-C error, 17-4
definitions, 7-4, 9-4
   creating, for SNAP-IX, 7-4
   creating, for SNAplus2, 9-4
   creating, for the gateway, 7-3, 9-3
   maintenance, 7-3, 9-3
   facilities, 1-17
   functions, 1-13
   LU, 8-4
   network, 7-4, 9-4
      name, in creating partner LUs, 7-11, 9-12
   profile, 11-14
   profile definitions, 8-3
   profiles, 7-2, 9-2
   protocol, 1-11, 7-6, 9-7
   remote access, 1-18
   sample configuration file, 6-4
   sample profile definitions, 8-3
   security, C-13
      option and IBM Communications Server for
         AIX, 6-8, 7-18, 8-9, 9-20
      validation, 15-4
   security validation, 6-6, 7-17, 8-7, 9-18
   SECURITY=PROGRAM, 6-7, 7-17, 8-8, 9-19
   SECURITY=SAME, 6-8, 7-18, 8-9, 9-20
   send/receive buffer, C-3
session allocation mode, DRDA_CMSRC_CM_IMMEDIATE, C-4
TPN, 8-6
SNA LU6.2
configuring DB2/400, 5-5
configuring DB2/OS390, 5-3
SNA node, 7-2, 7-4, 7-7, 9-2, 9-4, 9-8
configuring, 7-5, 9-5
SNA security option, 6-7, 7-17, 8-8, 9-19
SNA Server
Link Profile, 8-5
Local LU Profile, 8-4
Partner LU Location Profile, 8-5
Partner LU Profile, 8-5
profiles
creating, 8-3
sample profile definitions, 8-3
Side Information Profile, 8-5
SNA server
network requirements, 3-2
sna_domm.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, 7-4
definitions
for the gateway, 7-4
stored in directory, 7-3
Side Information Profiles required, 7-2
testing the connection, 7-19
SNAPplus2
configuring, 9-1
configuring, using xsnapadmin, 9-4
SNAPplus2
additional configuration needed for multiple
dependent LUs, 7-3, 9-3
configuring, using xsnapadmin program, 9-2
definitions
for the gateway, 9-4
stored in directory, 9-3
Side Information Profiles required, 9-2
testing the connection, 9-21
version required, 3-4
snapshots
known restrictions, 2-9
Oracle Snapshot feature, 13-9
software requirements, 3-3
Solaris
system support for APPC, 6-1
sort order
with ORDERBY, 14-25
SOUNDEX function, F-1
SPUFI, a database native tool, 11-11
SPX/IPX protocol, 1-4
SQL
ANSI standard, 1-7
clause compatibility, 14-26
clauses
DELETE, 14-34
INSERT, 14-34
SELECT WHERE, 14-34
UPDATE, 14-34
commands
describe, known restrictions, 2-6
constructs
Oracle processing, 14-8
differences in the gateway, 1-14
errors mapped to Oracle error codes, 17-5
functions
ABS, 14-25, F-1
ACOS, F-1
ADD_MONTHS, F-1
ASCII, 14-25, F-1
ASIN, F-1
ATAN, F-1
ATAN2, F-1
CEIL, 14-25, F-1
CHAR_TO_ROWID, F-1
CHR, 14-25, F-1
CONVERT, 14-25, F-1
COS, 14-25, F-1
COSH, 14-25, F-1
DECODE, 14-25, F-1
DUMP, 14-25, F-1
EXP, 14-25, F-1
FLOOR, 14-25, F-1
GREATEST, 14-25, F-1
GROUPBY, 14-25
HAVING, 14-25
HEXTORAW, 14-25, F-1
INITCAP, 14-25, F-2
INSTR, 14-25, F-2
INSTRB, 14-25, F-2
LAST_DAY, F-2
LEAST, 14-25, F-2
LENGTH, 14-25, F-2
LENGTHB, 14-25, F-2
LN, 14-25, F-2
LOG, 14-25, F-2
LOWER, 14-25, F-2
LPAD, 14-25, F-2
LTRIM, 14-25, F-2
MOD, 14-25, F-1
MONTHS_BETWEEN, F-1
NEW_TIME, F-1
NEXT_DAY, F-1
NLS_INITCAP, F-1
NLS_LOWER, F-1
NLS_UPPER, F-1
NLSSORT, 14-25, F-1
ORDERBY, 14-25
POWER, 14-25, F-1
RAWTOHEX, 14-25, F-1
REPLACE, 14-25, F-1
ROUND, 14-25, F-1
ROWIDTOCHAR, F-1
RPAD, 14-25, F-1
RTRIM, 14-25, F-1
SIGN, 14-25, F-1
SIN, 14-25, F-1
SINH, 14-25, F-1
SOUNDEX, F-1
SQR1, 14-25, F-2
STDDEV, 14-25, F-2
SUBSTR, 14-24, 14-25, F-2
SUBSTRB, 14-25, F-2
SYSDATE, F-2
TAN, 14-25, F-2
TANH, 14-25, F-2
TO_CHAR, F-2
TO_DATE, F-2
TO_LABEL, F-2
TO_MULTI_BYTE, F-2
TO_NUMBER, F-2
TO_SINGLE_BYTE, F-2
TRANSLATE, 14-25, F-2
TRUNC, 14-25, F-2
UID, F-2
UPPER, 14-25, F-2
USER, F-2
USERENV, F-2
VARIANCE, 14-25, F-2
VSIZE, F-2
WHERE, 14-25
functions and Native Semantics, 14-24
functions list, F-1
gateway architecture, 1-3
gateway transparency, 1-7
ISO standard, 1-7
native DB2, 1-7
passthrough, 14-36, 14-38
statements, 14-2, C-7
DB2, 14-36
issued through the gateway, 13-11
passing through gateway, 14-36
syntax, 14-35
tracing, not to be used in production
environment, 13-11
SQL functions
column functions, 14-8
compatible, defined, 14-8
compensated, defined, 14-9
DB2/400, 14-17
DB2/OS390, 14-10
DB2/UDB, 14-13
DB2/VM, 14-20
translated, defined, 14-8
with Native Semantics, 14-25
SQL tracing
in Oracle database, 17-8
in the gateway, 17-8
LOG_DESTINATION, 17-9
ORACLE_DRDA_TCTL parameter, 17-9
TRACE_LEVEL parameter, 17-9
SQL*Net
gateway migration problems, 16-3
Heterogeneous Services (HS) facility, 12-3
replaced by Oracle Net, 12-3
SQL*Plus
connecting to gateway, 12-8
COPY command, 13-10
DESCRIBE command, known restrictions, 2-6
extending gateway uses, 1-14
known restrictions, 2-6
moving data, 1-9
sqlstate, mapped errors, 17-5
SQRT SQL function, 14-25, F-2
stability, of cursor, C-7
Startup Shell Script
and gateway configuration, 11-4
backup and recovery of gateway configuration, 11-12
binding the DRDA gateway package, 11-8
choosing, 11-4
creating, 11-5
customizing, B-3
driver.ora file, 12-4
NLS parameters, D-8
parameter, tailoring to configure the host, 11-4
parameters governing UNIX and IBM Communications Server for AIX environments, 11-6
parameters new since V4 gateway, 16-3
tailoring file for IBM Communications Server for AIX, 11-6
troubleshooting environment, 11-4
startup time, gateway, 11-4
statements
CREATE DATABASE LINK, 13-2
DB2 CREATE TABLE, 1-7
DDL, 14-36
DELETE, 14-32, 14-36
DROP DATABASE LINK, 13-3
GRANT, 11-13
INSERT, 14-36
PREPARE TRANSACTION, 13-8
SELECT, 14-3, 14-9, 14-38
read-only gateway, 13-9
SQL
DB2, 14-36
JOIN, 14-2
SELECT, 14-38
UPDATE, 14-32, 14-36
STDDEV SQL function, 14-25, F-2
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
SNAplus2, 9-2
stored procedure
creating on DB2, 14-6
DB2, 14-7
native DB2, 1-8
Oracle and non-Oracle, 1-8
Oracle database server
PL/SQL, 14-3
remote instance, 14-3
Oracle, description, 1-8
Oracle9i Server
local instance, 14-3
using, 14-3
restriction, 2-7
usage, C-6
using DRDA server, 14-5
stored procedures, 1-5
DB2, 13-9, 14-5
DB2IND, G-2
GTW$_BIND_PKG, 11-8
ORAIND, G-4
REVISE_SALARY, 14-6
using with the gateway, 14-3
string concatenation
known restrictions, 2-7
string index, with Native Semantics, 14-24
Structured Query Language, also see SQL, 1-3
SUBSTR SQL function, 14-24, 14-25, F-2
known restrictions, 2-6
with Native Semantics, known restrictions, 2-6
SUBSTRB SQL function, 14-25, F-2
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, 1-13
feature, 13-5
for location transparency, 14-4
SYSDATE function, F-2
system privileges - BINDADD and BINDAGENT, 5-3, 5-7, 5-9

T
table
 - create a table in DB2, 14-37
 - insert a row into a DB2 table, 14-37
TABLE_PRIVILEGES view, A-15
tables
 - EXPLAIN_PLAN, 1-16
 - ORACLE2PC, 11-10, 11-12, 13-8
 - ORADRDA.ORACLE2PC, 13-8
TAN SQL function, 14-25, F-2
TANH SQL function, 14-25, F-2
TCP/IP
 - adapter, B-4
 - concurrent connections, 3-2
configuring
 - DB2/400, 5-5
 - DB2/OS390, 5-3
 - DB2/UDB, 5-7
 - DB2/VM, 5-9
configuring the host, 11-6
configuring under Solaris, 10-2
DRDA_CONNECT_PARM, C-4
facilities, 1-17
format, gateway initialization file parameter, C-4
functions, 1-13
modifying tnsnames.ora in Oracle Net, 12-5
network attachment, 3-2
number of database link connections, 14-8
protocol, 1-4, 1-6, 1-11
remote access, 1-18
security, 15-4
security validation, 15-4
support, 1-16
terminology defined, 1-10
tg4drda directory, also known as ORACLE_HOME, 4-5
TIME datatype, 14-29
time operations, 14-29
TIMESTAMP datatype, 14-29
TNSNAMES.ORA changes to, during migration problems, 16-3
extra parameter needed (HS=), 2-3
tnsnames.ora
 - connect descriptor, 13-2
 - file, 11-13, 11-14, 12-5
 - modifying for Oracle Net, 16-6
 - sample file, B-3
TO_CHAR function, F-2
TO_DATE function, 14-30, 14-32, 14-33, F-2
TO_DATE function main topic, 14-34
TO_LABEL SQL function, F-2
TO_MULTI_BYTE function, F-2
TO_NUMBER function, F-2
Token Ring, 3-2
token, package consistency, C-10
tools and the gateway, 1-14
TP name, 7-15, 9-16
TP_name
description, 6-5
TPN, 8-6
trace control, C-16
trace parameters setting, 17-8
TRACE_LEVEL parameter, 17-9, C-17
tracing, C-16, C-17
code, C-17
SQL statements, 13-11
trade-off, Native Semantics, 14-24
transaction mode, read-only, C-12
transaction program name, remote, 8-6
transactions
 - CICS, 14-7
 - IMS, 14-7
TRANSLATE SQL function, 14-25, F-2
translation, C-8
transparency, 1-4
access method, 1-4
data storage, 1-4
location, 1-4
native semantics, 14-24
network, 1-4
triggers
  for Oracle9i Server, 13-9
TRUNC function, F-2
TRUNC SQL function, 14-25
TSO
  configuring DB2/OS390, 5-3
two-phase commit
  ORACLE2PC table
    DB2/400, 5-5
    DB2/UDB, 5-7
    DB2/VM, 5-9
ORACLE2PC table, DB2/OS390, 5-3
processing transactions, 13-8
protection, 1-5
unsupported statement, 13-8

U
UID function, F-2
UNION
  capability, 1-5
  operator, 14-26
  SQL set operators and clauses, 14-26
UNION ALL
  operator, 14-26
  SQL set operators and clauses, 14-26
UNIX
  Bourne shell
  syntax, 11-4
UPDATE
  known restrictions, 2-9
  operation, 14-2
  read-only gateway, 13-9
  SQL clause, 14-34
  statement, 14-36
UPPER SQL function, 14-25, F-2
USER function, F-2
user ID mapping
  AVS, 15-6
  DB2/400, 15-7
  DB2/OS390, 15-5
  DB2/VM, 15-6
user ID translation
  DB2, 15-6
user privileges, 14-38
USER_CATALOG view, A-15
USER_COL_COMMENTS view, A-16
USER_CONS_COLUMNS view, A-17
USER_CONSTRAINTS view, A-16
USER_DB_LINKS data dictionary view, 13-3
USER_INDEXES view, A-17
USEROBJECTS view, A-19
USERSYNONYMS view, A-19
USER_TAB_COLUMNS view, A-22
USER_TAB_COMMENTS view, A-23
USER_TABLES view, A-20
USER_USERS view, A-23
USER_VIEWS view, A-24
USERENV function, F-2
USING clause, 13-2
USING parameter, modifying tnsnames.ora file, 12-5
Using the gateway, 13-1

V
VALUES clause
  functions not allowed by DB2, 14-34
VARCHAR datatype, 14-28
VARGRAPHIC datatype, 14-28
variable
  bind, 14-36
  input bind, 14-30
VARIANCE SQL function, 14-25, F-2
view
  creating, 11-13
  data dictionary
    emulation on DRDA server, 14-38
views
  catalog
    DB2/400, 14-39
    DB2/OS390, 14-39
    DB2/UDB, 14-39
    DB2/VM, 14-39
VM VTAM
  configuring, 5-9
VSAM
  file, 14-7
VSIZE function, F-2
VTAM
  configuring DB2/OS390, 5-3
  definitions
    relationship to SNAplus2 definitions, 7-19, 9-21
    same local LU name as on remote DRDA server, 7-10, 9-11
    security validation, 6-6, 7-17, 8-7, 9-18

W
WHERE clause, 14-26
  known restrictions, 2-9
WHERE CURRENT OF CURSOR clause
  known restrictions, 2-9
WHERE SQL function, 14-25
workarounds, 2-5

X
xsnaadmin
  creating SNA definitions, 7-4
  invoking, 7-4
  SNAP-IX definitions, 7-4
  tool to configure SNAP-IX, 7-2
  tool to create SNAP-IX definitions, 7-3
xsnapadmin, 9-4
  creating SNA definitions, 9-4
  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, 14-35