

Oracle® Transparent Gateway for DB2

Installation and User's Guide

Release 9.2.0.1.1 for OS/390

August 2002

Part No. A95217-01

ORACLE®

Oracle Transparent Gateway for DB2 Installation and User's Guide Release 9.2.0.1.1 for OS/390

Part No. A95217-01

Copyright © 1996, 2002, Oracle Corporation. All rights reserved.

Primary Author: Enterprise Platforms Division

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

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

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

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

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

Oracle is a registered trademark, and Oracle8, Oracle8i, Oracle9i, Oracle Discoverer, Oracle Net, Oracle Open Gateways, Oracle Transparent Gateway, PL/SQL, Pro*C, Pro*COBOL, SQL*Forms, SQL*Net and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xiii
Preface.....	xv
Documentation Accessibility	xix
Accessibility of Code Examples in Documentation.....	xix
1 Introduction	
Version 9 Gateways	1-2
Advantages of the Gateway.....	1-2
SQL	1-5
Stored Procedures.....	1-5
Applications	1-6
Two-Phase Commit and Multisite Transactions	1-7
Site Autonomy.....	1-7
Migration and Coexistence	1-7
Security	1-8
Protection of Current Investment.....	1-8
Gateway Architecture	1-8
How the Gateway Works.....	1-9
2 Release Information	
Product Set	2-2
Changes and Enhancements	2-2
Changes and Enhancements in Version 8.1.7.....	2-2

Changes and Enhancements in Version 9.2.0	2-2
Known Problems	2-3
Known Restrictions	2-4

3 System Requirements

CPU	3-2
Disk Space	3-2
Virtual Memory	3-2
Software Requirements	3-2
Operating System	3-3
UNIX System Services (USS).....	3-3
IBM Maintenance	3-3
Additional Software Requirements.....	3-3
Oracle Database Server Requirements	3-4
Oracle Net Requirements	3-4
Distribution Kit	3-5

4 Installation

Installation Checklists	4-2
Setup and Initialization Checklist	4-2
Installation Checklist.....	4-2
Post-installation Checklist	4-2
Setup and Initialization Steps	4-3
Step 1: Load the Oracle Installation JCL.....	4-3
Step 2: Create the ISPF Libraries.....	4-4
Step 3: Invoke the Product Installation and Customization Process.....	4-7
Step 4: Select Product Set from the Oracle Product Install Menu (Panel OR@PRIM)	4-8
Step 5: Select Options from the Oracle Primary Option Menu (Panel OR@INST)	4-8
Product Installation	4-10
Step 1: Modify Tape Unit and Library Index (Panel ORINIP00).....	4-10
Step 2: Select Oracle Products for Installation (Panel ORPRODS)	4-10
Step 3: Select Language Message Modules (Panel ORLANG).....	4-11
Step 4: INSTLIB/ISPSLIB File Tailoring Information (Panel ORINIP15).....	4-11
Step 5: Define JOB Cards for Install Jobs (Panel ORINIP20)	4-11
Step 6: Define PROCs, CLISTs, Linklist, and Temporary Space (Panel ORINIP25)	4-12

Step 7: Review /Modify Space Specifications for Major Libraries (Panel ORDSN).....	4-12
Step 8: Specify VOLSER for Other Libraries (Panel ORDSNO).....	4-12
Step 9: Product Installation Definition Complete (Panel ORINIP60)	4-13
Step 10: Generate the Installation Job	4-13
Step 11: Run the Installation Job (ORIJA01)	4-13
Step 12: Run the Generated Installation Jobs.....	4-13
Post-Installation Steps	4-14
Step 1: Define OSDI Subsystem for Oracle Gateways.....	4-15
Step 2: Associate Userids with Services	4-20
Step 3: Add Program Properties.....	4-21
Step 4: Authorize the Gateway Load Library.....	4-22
Step 5: Re-IPL OS/390 or Use Dynamic OS/390 Commands.....	4-22

5 Configuring the Gateway

Introduction	5-2
Checklists	5-2
Configuration Checklist	5-2
Post Configuration Checklist.....	5-3
Configuration Steps	5-3
Step 1: Configuring a Gateway Service Using ISPF Panels	5-3
Step 2: Run the sidJB00 Configuration Job.....	5-10
Step 3: Run the <i>sidJC01</i> and <i>sidJC02</i> Configuration Jobs	5-11
Step 4: Make Authorization and Local Date Exits Available to DB2.....	5-11
Step 5: Run the Scripts to Create Required Tables and Views in DB2.....	5-12
Step 6: Bind the DB2 Package	5-12
Step 7: Bind the DB2 Plan.....	5-15
Step 8: Grant EXECUTE on DB2 Plan.....	5-18
Step 9: Edit the PARMLIB Members	5-18
Step 10: Gateway Service Definition.....	5-29
Step 11: Setup Gateway JCL.....	5-31
Step 12: Start the Gateway.....	5-36
Post Configuration Steps	5-36
Step 1: Move Reentrant Modules to OS/390 Link Pack Areas	5-36
Step 2: Examine Oracle Dump Data Sets and Modify as Necessary	5-37
Step 3: Examine Oracle Trace Data Sets and Modify as Necessary	5-37

6 Oracle Net

Overview	6-2
OSDI Listener Architecture	6-2
OSDI Listener Filenames	6-3
Configuring the OSDI Listener	6-4
Network Service Definition.....	6-5
Service Name.....	6-5
TYPE	6-5
PROC	6-5
PARAM	6-5
OSDI Listener Region JCL	6-6
TCP/IP Network Considerations.....	6-8
Client-Server Access Using the OSDI Listener.....	6-9
Operating the OSDI Listener	6-10
Formatting OSDI Listener Trace Files	6-12
Oracle Advanced Security Option Encryption	6-12
Setting Up ASO Encryption for Test.....	6-12
Testing ASO Encryption	6-13

7 Administering the Gateway

Operation of the Gateway Subsystem with OSDI.....	7-2
Controlling Access to OSDI Subsystem Commands.....	7-2
Controlling Access to OSDI Services.....	7-4
Gateway Security	7-4
SAF Router Considerations	7-5
Gateway User Exit Facility	7-5
Specifying an Exit Module	7-6
DB2AUTH Exit	7-8
Sample Exit Programs	7-9

8 Using the Gateway

Database Link Behavior	8-1
Creating Database Links.....	8-2
Creating Database Links Using Oracle Net	8-2

Guidelines for Database Links.....	8-3
Accessing Data through Database Links.....	8-3
Dropping Database Links.....	8-3
Examining Available Database Links.....	8-3
Limiting the Number of Active Database Links	8-4
Managing Threads.....	8-4
KEEPALIVE	8-4
Canceling DB2 Threads	8-5
Gateway CPU Time	8-5
Using DB2 Cursors	8-5
Using the Synonym Feature.....	8-6
Read-Only Gateway	8-6
Performing Distributed Queries.....	8-7
Example of a Distributed Query	8-7
Two-Phase Commit Processing.....	8-8
Distributed DB2 Transactions.....	8-8
Replicating in a Heterogeneous Environment.....	8-8
Oracle Database Server Triggers	8-8
Oracle Snapshots	8-8
Copying Data from the Oracle Database Server to the DB2 Server	8-9
Triggers	8-9
SQL*Plus COPY Command	8-9
STREAMS Replication	8-10
Copying Data to the Oracle Database Server from the DB2 Server	8-14

9 Developing Applications

Gateway Appearance to Application Programs.....	9-2
Array Processing	9-2
Fetch Reblocking.....	9-3
Using Oracle Stored Procedures with the Gateway.....	9-3
Using DB2 Stored Procedures with the Gateway.....	9-4
Oracle Application DB2 Stored Procedure Completion	9-4
Procedural Feature Considerations with DB2.....	9-6
Passing DB2 SQL Statements through the Gateway	9-7
Using the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE Function	9-8

Retrieving Results Sets Through Passthrough	9-9
DB2 Data Types to Oracle Data Type Conversion	9-10
Performing Character String Operations	9-11
Converting Character String Data Types	9-12
Performing Date and Time Operations	9-12
DB2 Local Date Exit	9-13
Date Considerations in SQL Coding.....	9-14
NLS_DATE_FORMAT Support.....	9-15
Oracle TO_DATE Function	9-15
Date Arithmetic.....	9-16
Performing Numeric Data Type Operations	9-16
Oracle ROWID Column	9-17
Double Byte Character Set Support	9-17
CHAR FOR BIT DATA	9-18
SQL Functions	9-18
Oracle Database Server SQL Construct Processing	9-19
SELECT without the FOR UPDATE Clause	9-19
SELECT FOR UPDATE, INSERT, and DELETE Clauses.....	9-20
Oracle Database Server and DB2 Differences	9-20
Mass Delete from a Segmented Tablespace.....	9-20
Mapping the COUNT Function.....	9-21
Oracle Bind Variables.....	9-21
Oracle Data Dictionary Emulation on a DB2 Server	9-21
Using the Gateway Data Dictionary	9-22
DB2 Special Registers	9-22

10 Error Messages, Diagnosis, and Reporting

Message and Error Code Processing	10-2
Mapping DB2 Error Messages to Oracle Error Messages.....	10-2
Interpreting Message Formats	10-3
Messages Generated by Oracle Transparent Gateway for DB2.....	10-4
Examples	10-4
Diagnosing Errors Detected by the Oracle Database Server	10-5
Oracle Support Services	10-5
Providing Error Documentation	10-5

General Documentation Requirements	10-6
Error Diagnosis	10-7
Components	10-7
Error Categories	10-7
Documentation Errors	10-7
Incorrect Output	10-8
Oracle External Error	10-8
Abend	10-8
Program Loop	10-9
Performance.....	10-9
Missing Functionality	10-10
System Dumps	10-10
System Dump Data Sets	10-10
Operator Initiated Dumps.....	10-11
GTF	10-11

11 Migration and Coexistence with Existing Gateways

OSDI Differences	11-2
Summary of Changes.....	11-2
Configuring and Initializing an OSDI Subsystem.....	11-3
Configuring a Gateway Service	11-4
File Processing Considerations.....	11-5
Operating a Gateway Service	11-5
Oracle Net or Network Service.....	11-6
Migration and Upgrade	11-7
Release Incompatibilities	11-7
Migration and Upgrade Steps	11-8
Configuring Multiple OSDI Gateway Services	11-9
MPM/TNS and OSDI Coexistence	11-10

A Data Set Names and Space Allocations

Using the NEWDSRPT CLIST	A-2
Oracle Default Data Sets	A-2

B OSDI Subsystem Command Reference

Command Types and Processing	B-2
System Symbols in Commands	B-2
Definition Commands	B-3
Structures	B-3
Service Group Definition Commands	B-3
DEFINE	B-3
ALTER	B-4
SHOW	B-5
Service Definition Commands	B-5
DEFINE	B-5
ALTER	B-8
SHOW	B-10
Operating Commands	B-11
Available Commands	B-11
Commands	B-11
START	B-11
DISPLAY	B-12
DRAIN	B-13
RESUME	B-13
STOP	B-14
KILL	B-14
OSDI Command Keyword Abbreviations	B-15

C National Language Support

Overview	C-2
Default Character Set Changes	C-2
Oracle9i Enterprise Edition for OS/390	C-2
Gateway Configuration	C-3
NLS Parameters in <i>sid</i> ENV	C-3
DB2 Configuration	C-4
Supported Character Sets	C-4
Double-Byte Character Support	C-5
Supported Languages and Territories	C-7
Oracle Database Server and Client Configuration	C-8

Message Availability.....	C-8
---------------------------	-----

D The Oracle SMF Interface

Activating SMF Records.....	D-2
Specifying the Oracle Gateway Record Type.....	D-2
Starting SMF Recording.....	D-3
Stopping SMF Recording.....	D-3
Events Generating SMF Records.....	D-3
Interpreting an Oracle SMF Record	D-3
Contents of the SMF Header Section.....	D-4
Contents of the SMF Correlation Section.....	D-5
Contents of the SMF OSDI Data Section.....	D-6
Contents of the SMF Database Engine Data Section.....	D-7
Contents of the SMF Oracle Net Data Section.....	D-7
ORAFMT Sample Formatting Program	D-8

E Data Dictionary Views

ALL_CATALOG	E-2
ALL_COL_COMMENTS	E-2
ALL_CON_COLUMNS	E-3
ALL_CONSTRAINTS	E-3
ALL_IND_COLUMNS	E-4
ALL_INDEXES	E-4
ALL_OBJECTS	E-5
ALL_SYNONYMS	E-6
ALL_TAB_COLUMNS.....	E-6
ALL_TAB_COMMENTS.....	E-7
ALL_TABLES	E-7
ALL_USERS	E-8
ALL_VIEWS	E-9
COLUMN_PRIVILEGES.....	E-9
OTGREGISTER	E-9
TABLE_PRIVILEGES	E-10
USER_CATALOG	E-10
USER_COL_COMMENTS	E-11

USER_CONS_COLUMNS	E-11
USER_CONSTRAINTS	E-11
USER_INDEXES	E-12
USER_OBJECTS	E-13
USER_SYNONYMS	E-13
USER_TAB_COLUMNS	E-14
USER_TAB_COMMENTS	E-14
USER_TABLES	E-15
USER_USERS	E-16
USER_VIEWS	E-16

F Quick Reference to Oracle SQL Functions

G Sample Applications

DB2IND	G-1
ORAIND	G-3

H Installation Reference

Choosing Data Set Name Qualifiers	H-2
Data Set Name Qualifier Rules	H-3
OR@INST Panel Options	H-4

Index

Send Us Your Comments

**Oracle Transparent Gateway for DB2 Installation and User's Guide Release 9.2.0.1.1 for OS/390
Part No. A95217-01**

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.

Preface

This guide provides instructions for administering and configuring Oracle Transparent Gateway for DB2 for Release 9.2.0.1.1 for OS/390.

Intended Audience

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

- Installing and configuring the Oracle Transparent Gateway for DB2
- Administering the gateway
- Using the gateway

Understand the fundamentals of OS/390 operating systems before using this guide. This guide provides only information on Oracle products and their interactions with OS/390.

Product Name

The complete name for this product is Oracle Transparent Gateway for DB2 for OS/390. To maintain readability and conciseness in this document, the Oracle Transparent Gateway for DB2 for OS/390 may be referred to as the Gateway, the Oracle Transparent Gateway, the DB2 Gateway, or TG4DB2.

In all cases, the referenced product remains Oracle Transparent Gateway for DB2 for OS/390.

The Oracle Documentation Set

The documentation set has two parts: the documentation specific to the Oracle Transparent Gateway for DB2, and general gateway documentation. Your site automatically receives both documentation parts for the Oracle products that you have purchased. Use the general gateway documentation to learn about gateway concepts, and use the Oracle Transparent Gateway for DB2 documentation to learn how to install, administer, and use this gateway.

The Oracle Transparent Gateway for DB2 documentation consists of:

- *Oracle Transparent Gateway for DB2 Installation and User's Guide*
- *Oracle9i Enterprise Edition Messages Guide for OS/390*
- *Oracle9i Distributed Database Systems*

Related Documents

There are two parts to the documentation set: OS/390-specific documentation and product-specific documentation. Your site automatically receives both for the Oracle products you have purchased. Use the product-specific documentation to learn how to use a product, and use the OS/390-specific documentation to learn about special requirements or restrictions for using that product under OS/390.

OS/390-Specific Documentation

The OS/390-specific documentation set is used to install, maintain, and use Oracle9i for OS/390 products, and consists of:

- *Oracle9i Enterprise Edition Installation Guide for OS/390*
- *Oracle9i Enterprise Edition System Administration Guide for OS/390*
- *Oracle9i Enterprise Edition User's Guide for OS/390*
- *Oracle9i Enterprise Edition Messages Guide for OS/390*

Product-Specific Documentation

Product-specific documentation describes how to use the Oracle9i products. The information in these books is constant for all operating systems under which the products run. Refer to the environment-specific (in this case, the OS/390-specific) documentation for information that does not apply to every environment. Refer to the Oracle Technical Publications Catalog and Price Guide for a complete list of documentation provided for Oracle9i products.

SQL*Plus Prompts

The SQL*Plus prompt, SQL>, appears in SQL statement and SQL*Plus command examples. Enter your response at the prompt. Do not enter the text of the prompt (SQL>) in your response.

Storage Measurements

Storage measurements use the following abbreviations:

- K, for kilobyte, which equals 1,024 bytes
- M, for megabyte, which equals 1,048,576 bytes
- G, for gigabyte, which equals 1,073,741,824 bytes

Conventions Used in this Guide

Examples of input and output to the system are shown in a special font:

```
//SYSIN DSN=oran.orav.INSTJCL(member)
```

All output is shown as it actually appears. For input, the following conventions apply:

Uppercase indicates that a word or phrase must be entered exactly as spelled. For example: SYSIN

Italic indicates a variable or that a word or phrase must be substituted, such as the actual member name. For example: *member*

oran.orav is the standard example for high-level and second-level qualifiers. Substitute your system's actual high-level and second-level qualifiers.

<> **Angle brackets** indicate that the enclosed arguments are required and that at least one of the arguments must be entered. Do not enter the brackets themselves.

[] **Square brackets** indicate that the enclosed arguments are optional. Do not enter the brackets themselves.

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

| **Vertical lines** are used to separate choices.

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

Other punctuation must be entered as shown unless otherwise specified. For example, commas and quotes.

Commands, reserved words, and keywords appear in uppercase in both examples and text. A fileid can appear with both uppercase and lowercase text. When portions of a fileid appear in *italics*, the use of *italic characters* indicates that those portions can vary. Reserved words and keywords must always be entered as is, because they have reserved meanings within Oracle.

Switches

Switches are listed with a brief description and any OS/390-specific information. Refer to the product-specific documentation for syntax and for a thorough description of its purpose. Before using a command line option switch, such as -A or -C, refer to the appropriate product-specific documentation.

Documents Referenced in this Guide

Advanced Security Administrator's Guide

Oracle9i Administrator's Guide

Oracle9i Distributed Database Systems

Oracle9i Server SQL Reference

Oracle Net Services Book Set

OS/390 MVS Initialization and Tuning Reference

Programmer's Guide to the Oracle Call Interface

Programmer's Guide to the Oracle Precompilers

RACF Command Language Reference

RACF System Administrator's Guide

*SQL*Plus User's Guide and Reference*

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.

Introduction

In today's fast-paced global economy, information is a company's most valuable resource. Companies often have a variety of applications and data that are geographically scattered and use incompatible networks, platforms, and storage formats. In a single company, data is likely to be dispersed across several systems and hundreds of desktops. Oracle Open Gateways simplify complex systems and remove obstacles to information mobility.

The following topics are included:

- [Version 9 Gateways](#) on page 1-2
- [Two-Phase Commit and Multisite Transactions](#) on page 1-7
- [Site Autonomy](#) on page 1-7
- [Migration and Coexistence](#) on page 1-7
- [Security](#) on page 1-8
- [Protection of Current Investment](#) on page 1-8
- [Gateway Architecture](#) on page 1-8
- [How the Gateway Works](#) on page 1-9

Oracle Open Gateways use the facilities of the Oracle9i Heterogeneous Services. For further information on Heterogeneous Services, refer to *Oracle9i Distributed Database Systems*.

Version 9 Gateways

The Oracle9i database server continues to improve its distributed database facilities that make the integration of enterprise data practical. Data can be accessed remotely using both SQL and PL/SQL procedure calls in a transparent manner, as if the data were local. Also, data can be stored in both Oracle9i database servers and non-Oracle servers.

Version 9 gateways are tightly integrated with the Oracle9i database server, enabling improved performance and enhanced functionality while still providing transparent integration of Oracle and non-Oracle data. For example, connection initialization information is available in the local Oracle database server, reducing the number of round trips and the amount of data sent over the network. Running SQL 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 the enhancements in the Oracle9i database server, and you can quickly extend those benefits to your non-Oracle data.

Advantages of the Gateway

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

Oracle Transparent Gateway for DB2 gives you the power to integrate your heterogeneous system into a single, seamless environment, enabling 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 avoid the tedious, error-prone process of manual data transfer. Together with the Oracle tools, networking, and data server technology, Oracle Transparent Gateway for DB2 provides seamless, enterprise-wide information access.

Transparency at All Levels

By using Oracle Transparent Gateway for DB2, you can achieve transparency at every level within your enterprise.

- Location transparency

Users can access tables by name, without having to understand the physical location of the tables.

- Network transparency

The gateways exploit Oracle Net, enabling protocol independence. Regardless of the quantity or type of communication protocols used, data access is seamless. Protocols supported for Oracle Net include IBM TCP/IP High Performance Native Sockets (HPNS), and Oracle Net cross memory driver.
- 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.

Extends Database Services

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

- SQL functions

Your application can access all your data using Oracle SQL that is rich in features. Advanced Oracle9i database server functions, such as outer joins, are available, even if the target data stores do not support them in a native environment. The manner in which the gateways are integrated with the Oracle9i database server ensures 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 database server can utilize its advanced query optimization techniques to ensure SQL statements are run efficiently against any of your

data. The data distribution and storage characteristics of local and remote data are considered equally.

- Two-phase commit protection

The Oracle two-phase commit mechanism provides consistency across data stores by ensuring 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 are 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.

Extends Advanced Networking, Internet, and Intranet Support

The gateway integration with the Oracle9i database server extends to non-Oracle data the benefits of the Oracle internet and Oracle Net, and 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 the Oracle database server can also incorporate information from data stores accessible through the gateways. Web browsers can connect to the Oracle database server using any application server product that supports Oracle software.

- 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 the Advanced Networking Option (ANO).

- Wireless communication

Oracle Mobile Agents, an Oracle industry-leading mobile technology, enables wireless communication to Oracle9i database servers or any databases accessible through the gateways. This gives field personnel direct access to enterprise data from mobile laptop computers.

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 this information in a separate data dictionary stored in a hub. Applications access information using the hub dictionary instead of the native dictionaries of each database. This approach requires a great deal of manual configuration and maintenance. Administrators 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 DB2. The gateway uses the existing native dictionaries of each database. Your applications access data using the dictionaries designed specifically for each database, which means that 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 use native data control language (DCL) statements from an Oracle database server environment, allowing central administration of user privileges and access levels for heterogeneous data stores.

Passthrough and Native DB2 SQL

Running of native DB2 SQL can be passed through the gateway for running directly against DB2. This enables applications to send statements, such as a DB2 CREATE TABLE, to the gateway to run 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 and update 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 database 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 run DB2 stored procedures using standard Oracle PL/SQL. The Oracle application runs the DB2 stored procedure as if it were an Oracle remote procedure.

Applications

Any application or tool that supports the Oracle9i database server can access over 30 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 improvised query tools, web browsers, turnkey applications, and application development tools.

Oracle Developer

Use Oracle Developer to build applications that can manipulate data stored in your DB2 database and Oracle database server. This includes designing forms, producing graphic displays, and creating a wide range of objects.

Oracle Discoverer

Use Oracle Discoverer to analyze, manipulate, and copy data residing in your DB2 database. This product gives you access to corporate data using a powerful data analysis tool.

SQL*Plus

Use SQL*Plus for moving data between the databases. This product gives you the ability to copy data between your department databases and corporate Oracle database servers.

Oracle Database Server Technology and Tools

The gateway is integrated into the Oracle database 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 database server can be used to access heterogeneous data through the gateway.

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 implementing the gateway are the locking and transaction-handling capabilities of your target database.

Oracle Transparent Gateway for DB2, 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.

Site Autonomy

All Oracle database server products, including gateways, supply site autonomy. For example, administration of a data source remains the responsibility of the original system administrator. With site autonomy implemented, gateway products do not override the security methods of 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. As a result of this, the Oracle database 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.

Protection of Current Investment

With the gateway, you can continue to develop your information systems without foregoing your investments in current data and applications. Access your Oracle and DB2 data with a single set of applications and 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 want to migrate to Oracle database server technology and productivity, then the gateway allows you to control the pace of your migration. As you transfer applications from your previous technology to the Oracle database server, you can use the gateway to move the DB2 data into Oracle database servers.

Gateway Architecture

The gateway architecture consists of three main components.

1. Oracle Database Server

The Oracle database server is an Oracle instance functioning as a client to access the gateway.

2. Gateway

Oracle Transparent Gateway for DB2 must be installed on an OS/390 system.

3. DB2 server

The DB2 database server is the one being accessed by the gateway. IBM terminology for a DB2 server is DB2 Server for OS/390 or DB2 Universal Database (UDB) for OS/390.

Multiple Oracle database servers can access the same gateway. A single gateway installation can be configured to access one DB2 server.

How the Gateway Works

The gateway has no database functions. The gateway provides an interface by which the Oracle database server can direct SQL operations to a DB2 database.

Using a database link, the gateway is identified to the Oracle database server. The database link is the same construct used to identify other Oracle database servers.

Tables on the DB2 server are referenced in SQL as:

```
table_name@dblink_name
```

or

```
owner.table_name@dblink_name
```

If you create synonyms or views in the Oracle database server, then you can refer to tables on the DB2 server using simple names as though the table were local to the Oracle database server.

When the Oracle database server encounters a reference to a table on the DB2 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 DB2 server.

The gateway is responsible for sending these SQL statements to the DB2 server. The DB2 server is responsible for running the SQL statements and for fielding and returning responses.

Release Information

This chapter describes the changes and corrected problems in this release. The following topics are included:

- [Product Set](#) on page 2-2
- [Changes and Enhancements](#) on page 2-2
- [Known Problems](#) on page 2-3
- [Known Restrictions](#) on page 2-4

Product Set

The following table lists the versions of the components and utilities included on this tape.

Product	Release Number
Oracle Transparent Gateway for DB2	9.2.0.1.1
Oracle Net	9.2.0.1.1

Changes and Enhancements

This section lists the changes and enhancements to TG4DB2 by the release in which they were introduced.

Changes and Enhancements in Version 8.1.7

The following changes and enhancements were incorporated in the 8.1.7 release.

Auto Registration

Oracle Transparent Gateway for DB2 supports Heterogeneous Services auto registration. This feature will significantly speed session initialization by storing class capabilities in the server data dictionary rather than uploading the capabilities and storing them in memory for each session.

DB2 Version 6.1 and 7.1 Stored Procedures

Oracle Transparent Gateway for DB2 now supports IBM's new procedure for running Store Procedures in DB2 version 6.1 and 7.1.

Changes and Enhancements in Version 9.2.0

The following changes and enhancements were incorporated in the 9.2.0 release.

Operating System Dependent Interface (OSDI)

Like Oracle9i for OS/390, Version 9.2.0.1.1 of the Oracle Transparent Gateway for DB2 is based on OSDI (Operating System Dependent Interface), an execution environment for Oracle products on OS/390 which was introduced with Oracle8i for OS/390, Release 3 (8.1.7).

OSDI represents a significant change from the old subsystems in terms of how Oracle products interact with the OS/390 operating system. These changes do not generally affect Oracle product behavior and the interfaces used by customer applications, but they do affect the installation, configuration, and administration of these products on OS/390.

If you are already an Oracle for OS/390 customer with existing MPM or TNS subsystems, refer to [Chapter 11, "Migration and Coexistence with Existing Gateways"](#), for more information on differences between MPM and OSDI, and for migration and upgrade considerations.

TNSNAMES SID PARAMETER

Unlike previous MPM releases of the Oracle Transparent Gateway for DB2, the port number in the TNSNAMES entry is no longer sufficient to determine the destination of a connection request. The NET listener can now listen on a single port for multiple OS/390 destination address spaces. The TNSNAMES entry must now include a SID= parm. This parm value must match the SID(xxxx) value in the OSDI service definition. Please refer to ["Step 10: Gateway Service Definition"](#) on page 5-29 for more information on defining the SID= parm.

SQLplus Describe Support

Using RDBMS 9.2 as integrating server, you can now use the SQLplus Describe command to describe DB2 objects.

DB2 (var)graphic to Oracle (var)char(2) Mapping Support

This release of Oracle Transparent Gateway for DB2 provides DB2 (var)graphic to Oracle (var)char(2) mapping support. Refer to [Appendix C, "National Language Support"](#) for details.

Streams Heterogeneous Replication

Heterogeneous Replication is a new RDBMS feature utilizing Oracle gateway technology. You need a minimum release of 9.2.0.2 database server release to take advantage of the feature. Refer to Chapter 8, "STREAMS Replication" for details.

Known Problems

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

Attention: These problems are currently being addressed by the Oracle Corporation.

You should refer to the respective bugs for the current status of each problem.

Known Restrictions

The restrictions documented in this section are known to exist for the products included on this tape. Also refer to [Chapter 9, "Developing Applications"](#) for information or limitations when developing your applications.

IBM DB2 Release 5.1 and Above ASCII Tables

IBM DB2 release 5.1 and above supports ASCII and EBCDIC character sets. The character set selection is defined during table creation. Oracle Transparent Gateway for DB2 supports access to EBCDIC tables only. Tables created using the ASCII character set are not available to the gateway.

Owners of DB2 Components

DD Basic Tables and Views The owner of DD basic tables and views is OTGDB2. This cannot be changed.

Owner of DB2 Plan The qualifier for the DB2 plan must be the same as the owner for the ORACLE2PC table in the DD SQL script and it must be OTGDB2.

Binary Literal Notation

Oracle SQL uses hexadecimal digits surrounded by single quotes to express literal values being compared or inserted into columns defined as RAW. Currently, this is not converted to DB2 syntax (an X followed by quoted hexadecimal digits) when the SQL destination is the gateway. You must use bind variables to compare or insert into a DB2 server column defined with the FOR BIT DATA option.

Programmatic Limitations

Gateway design requires that all host variables in a SQL operation be bound before performing a describe function. When using the Oracle Call Interface (OCI), all OCI bind calls for a given statement must be completed before an OCI describe call is made.

Columns Defined with RAW Data

When you select RAW data into character bind variables, the CHAR column must be two times the size of the RAW data. Selecting RAW data into character bind variables causes implicit RAW to HEX conversion. If the character bind variable column is too small, then the SELECT statement fails.

Precompiler Limitations

The SQLCHECK option must be set to NONE when precompiling programs with the Oracle Precompilers.

SQL Limitations

Although most differences between the Oracle database server SQL and DB2 SQL are handled by the gateway, the following restrictions exist:

- Oracle ROWID is not supported

DB2 does not have a functional equivalent to Oracle ROWID. Tools or applications depending on Oracle ROWID are not supported.
- UPDATE and DELETE with the WHERE CURRENT OF CURSOR clause are not supported

When these statements are used in precompiler and PL/SQL programs, they rely internally on the Oracle ROWID function. Therefore, they are not supported.
- Oracle bind variables become DB2 parameter markers

Oracle bind variables become DB2 parameter markers when used with the gateway. Therefore, the bind variables are subject to the same restrictions as DB2 parameter markers. For example, the following statements are not allowed:

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

For more information about DB2 parameter marker restrictions, refer to the IBM documents for your platform and operating system.
- CONNECT_BY is not supported

The Oracle-specific CONNECT_BY clause is not supported.

Savepoints

If you try to use a savepoint in DB2, then you receive an ORA-2070 message. This error is appropriate because DB2 does not support savepoints.

GTWY_IDLE_TIME

The GTWY_IDLE_TIME option is no longer supported. You could still use OSDI KILL command manually to terminate the dedicated task which serves potentially long idle database link session. Refer to "[Operating Commands](#)" on page B-11 for details about OSDI commands.

System Requirements

This chapter lists the hardware and software requirements for installing the gateway. The following topics are included:

- [Software Requirements](#) on page 3-2
- [Oracle Database Server Requirements](#) on page 3-4
- [Oracle Net Requirements](#) on page 3-4
- [Distribution Kit](#) on page 3-5

Resource Requirements

The following Oracle Transparent Gateway for DB2 resource requirements are discussed in this chapter:

- CPU
- Disk space
- Virtual memory

CPU

The gateway requires any processor that can run the required OS/390 operating systems listed in "[Software Requirements](#)" on page 3-2.

Disk Space

Refer to [Appendix A, "Data Set Names and Space Allocations"](#), to determine the total Oracle non-database requirements, including code, JCL, and samples.

Virtual Memory

Each concurrent user of the gateway increases the virtual memory consumption, and it varies depending on:

- The number and complexity of SQL statements referring to tables through the gateway
- The number of columns and column sizes of those tables
- The configured network buffer size
- The number of open cursors and the HS_RPC_FETCH_SIZE initialization parameter

Oracle Corporation recommends defining 0M for the REGION parameter in the Oracle Transparent Gateway for DB2 address space.

Oracle Corporation also recommends defining 0M for the REGION parameter in the Oracle Net address space.

Software Requirements

Oracle Transparent Gateway for DB2 has requirements for the following:

- Operating system
- IBM maintenance
- Additional software
- Oracle database server software
 - Oracle9i database server requirements
 - Oracle Net requirements

The system software configuration described in the following 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

OS/390 V2R10 or higher is required.

UNIX System Services (USS)

An operational OS/390 UNIX System Services (USS) is required.

IBM Maintenance

Oracle Corporation recommends running Oracle Transparent Gateway for DB2 at the most current operating system service level. Apply the appropriate PTFs for the following IBM APARS:

APAR Number	Description
OW43816	Abend EC6 RSN083404P2 on socket() or accept() call from an SRB.
OW52769	STORAGE OBTAIN LOC=(EXPLICIT,64) CAUSES LOOP IN IGVSL0C. This occurs when an OS/390 system has greater than 2 GB of central storage.

Additional Software Requirements

The following software is required for installing the gateway:

Product	Use
ISPF	<p>IBM Interactive System Productivity Facility (ISPF) release 3.5 or later is required.</p> <p>IBM ISPF release 3.5 is used during the installation of the Oracle database server.</p> <p>If you are using a lower level, then Oracle Corporation recommends you change the text in your REXX EXECs to uppercase.</p> <p>This avoids problems with ISPEXEC functions and parameters.</p> <p>IBM ISPF version 4 is supported with the following restriction:</p> <p>For all panels to function correctly and match the documentation, set ISPF option 0.4 (Display Characteristics) <code>COMMAND LINE PLACEMENT ===> ASIS</code>, and do not split the screen.</p> <p>Splitting the screen can cause the CUA action line prompts to appear. <code>COMMAND LINE PLACEMENT ===> BOTTOM</code> can cause part of the dialog panel titles to be obscured.</p>
REXX	The IBM REXX product is used during the installation process.
IBM DB2	A version of IBM DB2 currently supported by IBM.

Oracle Database Server Requirements

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 Net Requirements

When you install Oracle Net, you have additional software and hardware requirements.

Oracle Net Protocol	Product	Required Protocol Release
Oracle Net IBM TCP/IP	OS/390	2.10 or higher

Oracle Net IBM TCP/IP

An OS/390 host using Oracle Net IBM TCP/IP requires any hardware capable of interfacing with the IBM TCP/IP stack.

The IBM TCP/IP software must be on the same OS/390 system as Oracle Net IBM TCP/IP.

Distribution Kit

Before installing the gateway, verify you have the correct tape and proper documentation:

- One Oracle Transparent Gateway for DB2 product tape
- Appropriate Oracle documentation for installing, administering, and using the gateway

The products needed for your gateway installation are distributed on one tape. The volume serial number of the tape is OS023A.

The data set names on this tape are in the format:

sysmod.Fxx

where:

sysmod is the SYSMOD name.

xx is a number from 1 to 25.

Installation

This chapter describes installing the Oracle Transparent Gateway for DB2. The following topics are included:

- [Installation Checklists](#) on page 4-2
- [Setup and Initialization Steps](#) on page 4-3
- [Product Installation](#) on page 4-10
- [Post-Installation Steps](#) on page 4-14

Each of these steps is described in this chapter. Installation messages are documented in the *Oracle9i Enterprise Edition Messages Guide for OS/390*.

Refer to [Chapter 11, "Migration and Coexistence with Existing Gateways"](#), for information about migration.

Installation Checklists

Use the following checklists for installation.

Setup and Initialization Checklist

- Step 1: Load the Oracle Installation JCL
- Step 2: Create the ISPF Libraries
- Step 3: Invoke the Product Installation and Customization Process
- Step 4: Select Product Set from the Oracle Product Install Menu (Panel OR@PRIM)
- Step 5: Select Options from the Oracle Primary Option Menu (Panel OR@INST)

Installation Checklist

- Step 1: Modify Tape Unit and Library Index (Panel ORINIP00)
- Step 2: Select Oracle Products for Installation (Panel ORPRODS)
- Step 3: Select Language Message Modules (Panel ORLANG)
- Step 4: INSTLIB/ISPSLIB File Tailoring Information (Panel ORINIP15)
- Step 5: Define JOB Cards for Install Jobs (Panel ORINIP20)
- Step 6: Define PROCs, CLISTs, Linklist, and Temporary Space (Panel ORINIP25)
- Step 7: Review/Modify Space Specifications for Major Libraries (Panel ORDSN)
- Step 8: Specify VOLSER for Other Libraries (Panel ORDSNO)
- Step 9: Product Installation Definition Complete (Panel ORINIP60)
- Step 10: Generate the Installation Job
- Step 11: Run the Installation Job (ORIJA01)
- Step 12: Run the Generated Installation Jobs

Post-installation Checklist

- Step 1: Define OSDI Subsystem for Oracle Gateways
- Step 2: Associate Userids with Services

- ❑ [Step 3: Add Program Properties](#)
- ❑ [Step 4: Authorize the Gateway Load Library](#)
- ❑ [Step 5: Re-IPL OS/390 or Use Dynamic OS/390 Commands](#)

Setup and Initialization Steps

The following steps guide you through the process of setting up your user environment to run the Oracle Installation Dialog Facility. If this is your initial installation of the Oracle Transparent Gateway for DB2, then proceed with [Step 1: Load the Oracle Installation JCL](#). If this release of the Oracle software is already installed, and if you would like to configure a Transparent Gateway for DB2, then refer to [Chapter 5, "Configuring the Gateway"](#).

Step 1: Load the Oracle Installation JCL

The Oracle Transparent Gateway for DB2 distribution tape is a standard-label tape containing many files, including the one that contains the JCL to start the Oracle installation. You can copy this file to a disk data set with the JCL provided here.

Before using the JCL, you must customize it for your site. The distribution tapes are designed for installations using 3380 type and 3390 type storage devices.

The following job loads the installation JCL:

```
//INSTAL1 JOB 1, 'GETFILE1', CLASS=A, MSGCLASS=X, NOTIFY=DBA1
//*
//S1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=ORACLE.OORX023.F1,
// DISP=OLD,
// UNIT=tape,
// VOL=SER=OS023A,
// LABEL=(1, SL, EXPDT=98000)
//SYSUT2 DD DSN=oran.orav.INSTLIB,
// DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA,
// VOL=SER=volser,
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920),
// SPACE=(27920, (600, 100, 20))
//SYSIN DD *
COPY INDD=SYSUT1, OUTDD=SYSUT2
SELECT MEMBER=OSPIJA00
/*
```

//

where:

<code>tape</code>	is the tape drive on which you mount the tapes.
<code>oran.orav</code>	are the high-level and second-level data set name qualifiers chosen for this installation.
<code>volser</code>	is the DASD volume serial number on which the Oracle INSTLIB library is to be allocated. The Oracle INSTLIB library contains installation-related material, including most of the jobs that you run during the installation process.

Step 2: Create the ISPF Libraries

The Partitioned Data Set (PDS), which is created when the first file is downloaded from the tape, contains member OSPIJA00. This member is a batch job that creates and loads (from the same tape) the Oracle ISPF libraries that are used during the Oracle installation. If these libraries already exist on your system, then they are deleted before the new data sets are created.

Member OSPIJA00 loads the following ISPF libraries:

ISPCLIB contains the Oracle installation CLISTS. This library also contains a CLIST called NEWDSRPT that produces a report listing the default library sizes for the data sets corresponding to the products that you selected during your installation. Refer to [Appendix A, "Data Set Names and Space Allocations"](#) for more information about using this CLIST.

Oracle Corporation requires you to unload the Oracle CLIST library into a data set with a record format (RECFM) of fixed block (FB) and a LRECL of 80 characters.

If you prefer to use variable block (VB) records for system CLISTS, then use ISPF option 3.3 to copy all of the library members into a VB data set with the same logical record length (LRECL) and block size as your existing CLIST libraries. Ensure that the new VB data set has the name *oran.orav.ISPCLIB*, where you substitute the high-level and second-level qualifiers of your installation for the *oran.orav* qualifiers in the ISPCLIB data set name.

ISPMLIB contains the Oracle installation messages.

ISPPLIB contains the Oracle installation panels.

ISPSLIB contains the Oracle installation skeletons.

ISPTLIB contains the Oracle installation tables.

In ISPF EDIT, modify the following items in the OSPIJA00 job to run on your system:

- jobname, accounting information, CLASS, and MSGCLASS on the JOB card
- symbolic parameters in the PROC statement (listed in [Table 4-1, "OSPIJA00 Job Parameters"](#) on page 4-6)

The last four symbolic parameters supply the data set names of existing ISPF libraries in your system. These parameters reference the associated DCB (data control block) attributes so that the created Oracle libraries can be assigned matching attributes. Ensure that the default data sets exist on your system. If they do not exist, replace the data set names with the correct names for your system.

If you are uncertain about the correct data set names in your system, issue the following TSO command from ISPF option 6 (TSO Command Processor) to view the current session allocations:

LISTALC STATUS

Use the first data set name that is listed above each of the following DDnames for each of the symbolic parameters in the job:

- ISPCLIB or SYSPROC
- ISPLIB
- ISPPLIB
- ISPSLIB

In the following example, the bold data set names are the ones that you specify in the symbolic parameters:

```
--DDNAME---DISP--
SYS1.USER.ISFPLIB
  ISPPLIB  KEEP
SYS2.USER.ISPLIB
  KEEP
TSO1.USER.ISPF.ISPLIB
  KEEP
SYS1.ISPF.CLIB
  SYSPROC  KEEP
SYS1.ISPF.ISPSLIB
  ISPSLIB  KEEP
SYS3.ISPF.MLIB
  ISPLIB   KEEP
SYS2.V21.ISPLIB
  KEEP
```

Table 4–1 OSPIJA00 Job Parameters

Parameter	Default	Description
INDEX	ORACLE.ORALIBV	High-level qualifiers for the four created data sets. These should normally be changed to the high-level and second-level qualifiers that you used in Step 1 for the INSTLIB data set.
TPUNIT	TAPE	Tape unit specification for the distribution tape
TPVOL	OS023A	Volume serial number specification for the distribution tape. Do not change this parameter.
PDASD	SYSDA	Disk unit specification for the created data sets

Parameter	Default	Description
PDVOL	XXXXXX	Volume serial number specification for the created data sets
TDASD	SYSDA	Temporary disk space unit name
DCBPLIB	ISP.SISPPENU	DCB data set model of the first ISPF panel library
DCBMLIB	ISP.SISPMENU	DCB data set model of the first ISPF message library
DCBSLIB	ISP.SISPSENU	DCB data set model of the first ISPF skeleton library
DCBCLIB	ISP.SISPCLIB	DCB data set model of the first ISPF CLIST library

When you have completed your changes to member OSPIJA00, save and submit the job. OS/390 requests a tape mount for the Oracle distribution tape on the device that is specified in the TPUNIT symbolic parameter. When the job ends, examine the output to confirm successful execution before proceeding to [Step 3: Invoke the Product Installation and Customization Process](#).

Step 3: Invoke the Product Installation and Customization Process

After you successfully create the ISPF installation libraries, invoke ORIPO01 to start the customization process.

To invoke the ORIPO01 CLIST, logon to TSO, enter ISPF, and select option 6 (TSO Command Processor) from the main ISPF menu. From option 6, use the EXEC command to invoke the ORIPO01 CLIST from the ISPCLIB data set that was created in "[Step 2: Create the ISPF Libraries](#)":

```
----- ISPF COMMAND SHELL -----
Enter TSO or Workstation commands below:
===> EXEC 'oran.orav.ISPCLIB(ORIPO01)'
```

Substitute your high-level and second-level qualifiers for *oran.orav* in the following examples.

During the CLIST execution, you are prompted three times:

1. At the first prompt, enter the high-level and second-level data set name qualifiers for the ISPF data sets that were copied from the tape. These values should match those that you specified in the INDEX parameter in "[Step 2: Create the ISPF Libraries](#)".

2. At the second prompt, enter C to confirm the qualifiers you entered.

After your second reply, you are prompted with ***. When you press [Enter], panel OR@PRIM is displayed.

Step 4: Select Product Set from the Oracle Product Install Menu (Panel OR@PRIM)

Place an S in front of Oracle Transparent Gateway for DB2 and enter the high-level and second-level qualifiers that you want used for this product set, for example, *ORAN.ORAV*. Generally, these qualifiers match the corresponding qualifiers used in Step 1 of the Set Up and Initialization. Refer to ["Data Set Name Qualifier Rules"](#) in [Appendix H, "Installation Reference"](#) for more information.

When you select Oracle Transparent Gateway for DB2 on panel OR@PRIM and press [Enter], panel OR@INST is displayed. You are led through a series of panels that let you select and customize the Oracle Gateway for your installation.

Step 5: Select Options from the Oracle Primary Option Menu (Panel OR@INST)

Different installation and customization options may be performed from this panel. In general, they should be performed in the order shown. However, you may choose not to perform some of the customization options. You select the option that you want to perform by entering the option number on the panel command line.

These options go in pairs, (1) a definition option which takes you through a series of ISPF panels and (2) a generate option which produces a job stream based on the information provided on the ISPF panels, to further populate the Oracle INSTLIB library. (The generate option is typically followed by other tasks to complete the installation or customization that you are performing.)

From the Oracle Primary Option Menu, select the task that you want to perform:

1. Define Oracle Products Installation Parameters

This is the option that will begin the product installation process. Because this is the process that actually unloads the Oracle products from the distribution tapes, it must be performed before any of the other options.

Go to the ["Product Installation"](#) section on page 4-10 for product installation instructions.

2. Generate Installation Job "ORIJA01"

Go to the ["Product Installation"](#) section on page 4-10 for instructions on when to select this option.

3. Define New Oracle Database Parameters

This is the option that will begin the customization process for a new Oracle database. Refer to the *Oracle9i Enterprise Edition System Administration Guide for OS/390* for information on how to complete the database customization process.

This option is not valid for a Transparent Gateway installation.

4. Generate Database Job "ORDJA01"

Refer to the *Oracle9i Enterprise Edition System Administration Guide for OS/390* for instructions on when to select this option.

This option is not valid for a Transparent Gateway installation.

5. Define New Gateway for DB2 Parameters

This is the option that will begin the customization process for a new Transparent Gateway for DB2 instance. Refer to [Chapter 5, "Configuring the Gateway"](#), for information on how to complete the gateway customization process.

6. Generate Gateway Job "ORGDJA01"

Refer to [Chapter 5, "Configuring the Gateway"](#), for instructions on when to select this option.

9. Reset all Product and Language Selections

This option relates only to option 1. It can be used to nullify all product and language selections made by the user on panels ORPRODS and ORLANG.

Refer to [Appendix H, "Installation Reference"](#), for more information about option 9.

X. Exit ORACLE Install Dialog facility.

This option, as well as PF3, may be used to terminate the Oracle Install Dialog facility.

Product Installation

After selecting the product installation option from the Primary Option Menu, you will see panel ORPRIM0, Define Installation Parameters. This panel shows a list of the various installation parameters that must be provided in order to complete the product installation. Press [ENTER] on the panel to begin specifying the installation parameters. You will be presented with the customization panels for the installation of the Oracle Transparent Gateway for DB2 software.

The following rules apply to the customization panels:

- Most customization parameters have default values that you can change by typing in a new value in place of the default value. Other required parameters have no default values, and you must supply values for these parameters before proceeding to the next panel.
- You can move forward from one panel to the next by entering C on the command line.
- Most panels allow you to return to the previous panel by pressing the [PF3] key.

Step 1: Modify Tape Unit and Library Index (Panel ORINIP00)

1. Oracle Distribution Library

Identify the OS/390 tape unit name on which you intend to mount the Oracle distribution tape during the installation jobs. The tape volume serial number that appears on this panel should not be changed unless you are instructed to do so by Oracle Support Services.

2. Oracle Library Index

Identify the high-level and second-level data set qualifiers to be used for the Oracle product libraries. The default values for these come from the information that you supplied on panel OR@PRIM. Enter new names if you want to change the default values.

When the information on this panel is correct, enter C to continue to the next panel.

Step 2: Select Oracle Products for Installation (Panel ORPRODS)

This panel shows a list of all the products on the distribution tape. Enter S next to the products that you want to install. All product prerequisites are chosen automatically. If you select a product that has optional products associated with it (which were not already selected), then you will see this panel displayed a second time with the optional product list.

When the information in this panel is correct, enter C to continue to the next panel.

Step 3: Select Language Message Modules (Panel ORLANG)

This panel shows a list of the National Language Support (NLS) language message modules available with this release. Enter S next to the modules that you want to install. American English is the standard message module set provided with Oracle Transparent Gateway for DB2 for OS/390 and is always installed. The more language module sets that you install, the greater your installation disk requirements are.

When the information in this panel is correct, enter C to continue to the next panel.

Step 4: INSTLIB/ISPSLIB File Tailoring Information (Panel ORINIP15)

1. Oracle ISPSLIB (skeleton) Library member

This item is showing the file tailoring skeleton member that will be used during the generate step. You cannot change this value.

2. Oracle Installation Library

Provide information about the INSTLIB library in which the installation jobs are built. This is normally the same data set that is created when the first file is downloaded from the Oracle distribution tape (in "[Setup and Initialization Steps](#)" step 1). However, you may change the Oracle INSTLIB library name if required. Enter **NEW** or **SHR** for DATA SET DISPOSITION. The DATA SET DISPOSITION defaults to SHR. If the data set is new, then specify a VOLUME SERIAL and DEVICE TYPE.

When the information in this panel is correct, enter C to continue to the next panel.

Step 5: Define JOB Cards for Install Jobs (Panel ORINIP20)

This panel allows you to define the JOB card structure that will be used with each of the generated installation jobs. The additional lines can be used to add JES subsystem control cards or similar requirements. The JOBNAME parameter and the NOTIFY parameter default to your TSO logon userid. Change the default information as needed.

When the information in this panel is correct, enter C to continue to the next panel.

Step 6: Define PROCs, CLISTs, Linklist, and Temporary Space (Panel ORINIP25)

1. Define Oracle Procedure and CLIST Target Libraries

Provide the data set names for the target libraries where the Oracle JCL procedures (cataloged procedures) and TSO CLISTs will reside. You also provide a suffix for the Oracle JCL procedures. Because the suffix is appended to the end of each installation procedure, you can load the new version of Oracle JCL procedures to the same target library as a prior release without replacing the older procedures. The suffix is required and defaults to XX.

2. Define Oracle Linklist Library

Several modules in the Oracle product set must reside in the system linklist. These modules take approximately 64K bytes of space. Provide the name of the system linklist library that you want these modules to be copied into.

3. Oracle Temporary Disk Space Unit Name

Provide the unit name for the temporary disk work space. The default, SYSDA, is acceptable in most systems.

When the information in this panel is correct, enter C to continue to the next panel.

Step 7: Review/Modify Space Specifications for Major Libraries (Panel ORDSN)

Provide the volume serial number, device type, SMS classes, and space information for each of the major Oracle library data sets listed on this panel. You should discuss your SMS configuration with your system programmer to determine appropriate values for this panel.

Refer to [Appendix A, "Data Set Names and Space Allocations"](#), for information about the Oracle data sets, their space allocations, and the NEWDSRPT CLIST, which produces a report listing default library sizes. The default space figures in this panel are for the entire product set. These allocations are too large for a product subset.

When the information in this panel is correct, enter C to continue to the next panel.

Step 8: Specify VOLSER for Other Libraries (Panel ORDSNO)

Provide the volume serial number, device type, and/or SMS classes for all other Oracle library data sets. The information specified in this panel applies to all other Oracle data sets, in other words, those not listed on panel ORDSN. You should discuss your SMS configuration with your system programmer to determine appropriate values for this panel.

When the information in this panel is correct, enter C to continue to the next panel.

Step 9: Product Installation Definition Complete (Panel ORINIP60)

The appearance of this panel indicates that you have completed the Product Installation Definition process. You may use [PF3] to back up through the previous panels to make any corrections that you need. If you are satisfied with all of the parameter values in the customization panels, you may use [PF4] to return to the Primary Option Menu panel.

Step 10: Generate the Installation Job

Select option 2 (Generate Installation Job) from the Oracle Primary Option Menu panel. The installation CLISTs generate a job stream in a new member, ORIJA01, in the Oracle INSTLIB library.

The output displays as the generate step is executing. When you see the GENERATION PROCESS COMPLETED message, press [Enter] to return to the Oracle Primary Option Menu. Then press [PF3] to exit the install dialog.

Step 11: Run the Installation Job (ORIJA01)

You do not need to edit the generated job, ORIJA01, although you might want to change the jobname. Submit this job after you have made any changes appropriate for your site.

ORIJA01 creates multiple members in the INSTLIB library, including a series of installation jobs with member names ORIJB00 through ORIJC00.

Examine the output from the ORIJA01 job to confirm its successful execution. The return code from ORIJA01 should be 0, indicating successful execution.

Step 12: Run the Generated Installation Jobs

Submit each job in the order shown below. Some jobs might not be present in your installation depending on the installation options that you selected.

Caution: Even if the basic JOB card and related data are specified properly in the associated customization panel, you might need to change information such as the jobname in each job before submitting it for execution.

Before running these jobs, ensure that the Oracle PROCLIB and TSO CLIST libraries that were specified in panel ORINIP25 already exist. Otherwise, you receive JCL errors.

The following list describes the purpose of each job and any special considerations for its execution.

ORIB00

This job deletes the Oracle distribution libraries (if any), reallocates new ones, and downloads the libraries for the selected products from the distribution tape. This job requires that the Oracle distribution tape be mounted.

The delete, reallocate, and copy operations are performed only for those data sets corresponding to the products that you selected to install.

ORIC00

This job copies several modules from the Oracle AUTHLOAD library to the system linklist library that you specified on panel ORINIP25. These modules take approximately 64K bytes of space.

After this job has been successfully run, if you plan to start your OSDI subsystem before the next IPL, then you must refresh the system linklist by issuing the following system operator command:

```
F LLA, REFRESH
```

After issuing the foregoing command, message CSV210I should appear after a brief delay.

This completes the Oracle Product Installation process. Refer to [Chapter 5, "Configuring the Gateway"](#), and [Chapter 6, "Oracle Net"](#), for information on how to configure the products installed.

Post-Installation Steps

When installing your Oracle gateway, you must perform the following pre-installation steps. These changes must be made and the OS/390 system IPLed before starting the gateway.

[Step 1: Define OSDI Subsystem for Oracle Gateways](#)

[Step 2: Associate Userids with Services](#)

[Step 3: Add Program Properties](#)

[Step 4: Authorize the Gateway Load Library](#)

[Step 5: Re-IPL OS/390 or Use Dynamic OS/390 Commands](#)

Step 1: Define OSDI Subsystem for Oracle Gateways

To run any Oracle products under OSDI, you must first create an OS/390 subsystem. The subsystem provides the internal interfaces that OSDI uses to process operator commands, to manage the execution of gateway and network services, and to manage connections between address spaces. One OSDI subsystem can support any number of distinct Oracle gateway instances, database instances, and network services.

Choosing a Subsystem Name and Command Prefix

The subsystem name is a 1-character to 4-character identifier. OS/390 requires that subsystem names begin with an alphabetic or national character, while subsequent characters can be alphanumeric or national. The name must be unique.

The subsystem requires a unique character string prefix to distinguish commands issued to it from system consoles and other system command sources. By default, OSDI will use the subsystem name as the command prefix. You can override this default if you wish, and specify a special character or a different alphanumeric string. Any character string prefix that you choose must not duplicate, or be a leading subset of, the command prefix of any other subsystem. Also, the character string prefix must not match any native OS/390 system command name or abbreviation. Oracle Corporation recommends using the subsystem name for the command prefix.

If you are already an OSDI database server customer, you can use the same OSDI subsystem to manage the Oracle Transparent Gateway for DB2 service. Refer to ["Service Definition Commands"](#) on page B-5, for syntax of the DEFINE SERVICE command.

The Subsystem Parameter File

During initialization, the subsystem opens and reads a sequential data set to obtain bootstrap initialization parameters and, optionally, any OSDI commands that are to be issued immediately after the subsystem is initialized. The file is usually a member of a PDS, and is created, viewed, and edited with TSO ISPF or a similar tool. It can have either fixed-length or variable-length records. If fixed-length records are used, then sequence numbers in the right-most record positions (for example, columns 73-80 of 80-byte records) are ignored.

Oracle Corporation recommends creating a PDS specifically for OSDI parameters and using it for subsystem parameters as well as for parameter files that are used by other components. Ten tracks of primary disk space, three tracks of secondary space, and eight directory blocks should be sufficient for most installations. This data set must be accessible for the subsystem to initialize, so it should not be subject to HSM migration or similar involuntary moves.

Only a single record, called the INIT record, is required in the subsystem initialization file. The INIT record has the following format:

```
INIT (ORASSI [, cmd-prefix] [, cmd-class] [, bind-class])
```

The INIT record has the following keywords and arguments:

- **INIT**

The word INIT can begin in position 1 or can be preceded by one or more blanks. It must be followed by one or more blanks and then by one to four positional parameters separated by commas and enclosed in a single pair of parentheses. The parentheses are required even if only the first positional parameter is coded. Do not include blanks in or among the positional parameters. If an optional positional parameter is not used, but a following one is, then a comma must be included for the unused parameter.

- **First positional parameter (ORASSI)**

The first positional parameter is the name of the OSDI subsystem load module that was copied to a system linklist library during installation. This will normally be ORASSI. This parameter is required.

- **Second positional parameter (cmd-prefix)**

The second positional parameter is the command prefix for the subsystem. Any characters that are legal in a command prefix (except the comma and the left or right parenthesis) can be included. Do not enclose the prefix in apostrophes or quotes unless those are part of the prefix. The command prefix can be up to 8 characters in length. If you omit this parameter, then the command prefix is assumed to be the subsystem name.

- **Third positional parameter (cmd-class)**

The third positional parameter is the System Authorization Facility (SAF) resource class name that is to be used when authorizing access to OSDI commands. This parameter should be specified if you have created a specific resource class for command authorization as discussed in the "Creating or Activating a Resource Class" section of the "Pre-Installation Activities" chapter

of the *Oracle9i Enterprise Edition Installation Guide for OS/390*. If you omit this parameter, then OSDI uses the FACILITY class for command authorization requests.

- Fourth positional parameter (`bind-class`)

The fourth positional parameter is the SAF resource class name that is to be used when authorizing client address spaces during bind to a service. This parameter should only be specified if you have created a specific resource class for bind. If you omit this parameter, then OSDI uses the FACILITY class for bind and connect authorization requests.

The following INIT record provides an example in which the SAF resource class names have been specified while the command prefix is allowed to default:

```
INIT (ORASSI, , $ORACMD, $ORACONN)
```

OSDI Commands in the Subsystem Parameter File

The INIT record is the only item that is required to be in the subsystem initialization file. As a convenience, however, you can include OSDI commands in the initialization file immediately after the INIT record. Usually, you will have a number of OSDI DEFINE commands there. You might also have OSDI START commands for any services that you always want started as soon as possible during an IPL. Oracle Corporation recommends that you place into the parameter file both the DEFINE SERVICEGROUP command and the DEFINE SERVICE commands for commonly-used services.

The purpose and complete syntax of all OSDI commands is covered in [Appendix B, "OSDI Subsystem Command Reference"](#). You should consider the following items regarding commands that are supplied in the subsystem parameter file:

- Do not include the subsystem command prefix. All commands in the subsystem parameter file apply to the subsystem which is being initialized and is reading the file.
- Each command must begin on a new record. The command verb can begin in position 1 or it can be preceded by blanks.
- A command can be continued by including a hyphen (or minus sign) as the last non-blank character on a record (excluding any sequence number). The continuation can begin in position 1 of the next record or it can be preceded by blanks. You cannot continue (split across records) a command parameter that is enclosed in apostrophes. The continuation hyphen is not interpreted as part of the command.

As commands in the parameter file are processed by the subsystem, the command images are not displayed on the console log, but the subsystem response messages appear as if the commands were entered at a console. Refer to ["Step 10: Gateway Service Definition"](#) on page 5-29 for more information on defining the gateway service.

Initializing the Subsystem

Before attempting to initialize the subsystem, the required OSDI subsystem modules must reside in the system linklist. Refer to job [ORIJC00](#) on page 4-14 for information about copying modules to the Linklist Library.

The subsystem can be initialized in either of two ways:

1. At system IPL, based on an entry that you add to the IEFSSNxx member of SYS1.PARMLIB.
2. At any other time by using the SETSSI ADD system operator command.

Oracle Corporation recommends that you add regularly-used subsystems to the IEFSSNxx member so that they are initialized correctly and automatically at every IPL. Use the SETSSI ADD system operator command when necessary, such as when you first install OSDI and want to try bringing up a subsystem without having to IPL.

Because OSDI uses OS/390 dynamic subsystem interfaces, the IEFSSNxx entry for an OSDI subsystem must use the newer keyword parameter format, not the old positional format.

Assuming that you have chosen G4XX as your OSDI subsystem name and that the subsystem parameter file is member SSP01 of the data set *ORAN.ORAV.PARMLIB*, an appropriate IEFSSNxx entry would be similar to the following:

```
SUBSYS SUBNAME(G4XX) INITRTN(ORASSINI)
      INITPARM('ORAN.ORAV.PARMLIB(SSP01)')
```

In the above example, ORASSINI is the name of the subsystem's initialization routine. This module was copied to a system linklist library during OSDI installation. It must be specified exactly as shown.

The subsystem parameter string (INITPARM keyword) must specify the data set name and, if applicable, the member name of the subsystem parameter file containing the INIT record and optional commands. If no member name is supplied, then this must be a sequential (DSORG=PS) data set.

After updating IEFSSNxx, an IPL is necessary to process the added entry. If you do not want to wait for an IPL, or if other circumstances exist in which a subsystem must be created without an IPL, then you can use a SETSSI ADD system operator command equivalent to the following:

```
SETSSI ADD,S=G4XX,I=ORASSINI,P='ORAN.ORAV.PARMLIB(SSP01)'
```

This example uses the minimal keyword abbreviations. The longer forms, SUBNAME (for S), INITRTN (for I), and INITPARM (for P), can be used if desired.

Caution: Oracle Corporation recommends that you use caution when entering the SETSSI ADD command. If you make a mistake in the parameter string (such as misspelling the data set or member name), then the subsystem will fail to initialize, and the subsystem name that you specified will be unusable until the next IPL.

Examples

The following code represents the contents of a subsystem initialization file that includes OSDI commands to define the service group and several services. The file ends with an OSDI SHOW command that will display the definitions in the system log.

```
INIT (ORASSI,G4XX)
DEFINE SERVICEGROUP SSID(G4XX) -
  DESC('Oracle TG4DB2 OSDI Subsystem')
DEFINE SERVICE ORAGW1 DESC('Oracle TG4DB2 Gateway') -
  TYPE(GTW) PROC(ORAGW01) -
  PARM('ORAN.ORAV.PARMLIB(ORAGW1)')
DEFINE SERVICE ORANET DESC('Oracle Net') -
  TYPE(NET) PROC(ORANET01) -
  PARM('HPNS PORT(1521)')
SHOW SERVICEGROUP LONG
```

Consider this file to be member SSP01 of ORAN.ORAV.PARMLIB as in the earlier examples, and assume that the subsystem will be initialized with a SETSSI ADD command using subsystem name G4XX. The OS/390 system log that results from the SETSSI ADD command would be similar to the following:

```
SETSSI ADD,S=G4XX,I=ORASSINI,P='ORAN.ORAV.PARMLIB(SSP01) '
IEF196I IEF237I 0C9A ALLOCATED TO SYS00049
MIS0012I INITIALIZATION OF ORACLE SUBSYSTEM ORSS COMPLETE 500
MIS0196I Service group G4XX defined 501
MIS0198I Service ORAGW1 defined 502
```

```
MIS0198I Service ORANET defined 503
MIS0195I Service group G4XX (Oracle OSDI Subsystem) 505
          Mode=*SYS , Systems=*ALL
          Service ORANET Type NET (Oracle Net)
          Service ORAGW1 Type ORA (Oracle TG4DB2 Gateway)
IEF196I IEF285I ORAN.ORAV.PARMLIB KEPT
IEF196I IEF285I VOL SER NOS= WHODAT.
IEFJ022I SETSSI ADD COMMAND FOR SUBSYSTEM G4XX COMPLETED 508
SUCCESSFULLY
```

Step 2: Associate Userids with Services

Oracle9i-managed services execute as system address spaces, similar to **started tasks** or STCs. Some of the OS/390 system functions that are invoked by Oracle9i services perform authorization checks based on the OS/390 userid that is associated with the service address space. Depending on the security configuration and standards of your installation, those system functions may fail if no userid is associated with the address space. You, or security personnel for your installation, may need to take steps to ensure that Oracle9i services have an associated userid that can be authorized for system functions that are invoked by the database and network services.

With RACF, this authorization is normally accomplished by defining appropriate profiles in the STARTED resource class. Each profile associates a RACF userid with a started task based on the JCL procedure name of the started task. You will choose and specify JCL procedure names for Oracle9i services when those services are configured. You may want to decide on procedure names now, however, so that RACF profiles can be defined. There are no special procedure naming requirements as far as Oracle9i is concerned, so you can choose procedure names that meet the standards or requirements of your installation. Of course, the names should not be the same as the names of any members already in your system procedure library. The ISPF panel-driven gateway configuration process will generate a JCL procedure name based on the service SID specified. The generated name is *G4sid*, where *sid* is the service SID. After the configuration process is complete, you may change the name of this procedure if it does not match the standard naming conventions at your site.

Defining the JCL procedure name in the USER resource class is an alternate method in which the procedure name itself is also used as the userid.

If you are already running the OSDI and TNS programs as started tasks (as opposed to submitting them as batch jobs), then your installation probably already has STARTED or USER profiles for the associated JCL procedures. You should not rely

on those for Oracle9i because the Oracle9i procedures should have different names. Plan to create at least two new STARTED or USER profiles, one for the gateway service and one for the network service. These may be all that you need, because different instances of a type of service can generally share the same JCL procedure. You may want to create additional profiles, though, if you want different instances of a service to run with different userids. Note that this requires using distinct JCL procedures even though the procedures themselves may be otherwise identical.

Details on the STARTED and USER resource classes are in the RACF System Administration Guide. The RDEFINE command that is used to add profiles is described in the *RACF Command Language Reference*.

With RACF, it is possible to associate a userid with a started task using a **started procedures table** that is built with Assembler macros somewhat like the resource class table discussed in the previous section. Activating such changes requires an IPL, however, and is not the preferred method. Refer to the *RACF System Administrator's Guide* for more information.

Step 3: Add Program Properties

The gateway and network service region programs must run nonswappable and noncancelable, and should not be subject to system time limits. In addition, the gateway service runs in protect key 7. These attributes are indicated by adding entries for these programs to the OS/390 Program Properties Table (PPT), via a member of the SYS1.PARMLIB data set named SCHEDxx, where xx is a 2-letter or 2-digit suffix. You may need to work with your systems programmer to determine the correct member name and to add the entries. The entries that you add should be similar to those in the following example. The comments, which are included for clarity, are allowed but are not required.

```

/* SCHEDxx PPT entry for Oracle gateway region */
PPT PGMNAME(ORARASC) /* Program (module) name */
NOCANCEL /* Not cancelable */
KEY(7) /* Protection key */
NOSWAP /* Not Swappable */
SYST /* Not subject to timing */
/* SCHEDxx PPT entry for Oracle network region */
PPT PGMNAME(ORANET) /* Program (module) name */
NOCANCEL /* Not cancelable */
NOSWAP /* Not swappable */
SYST /* Not subject to timing */

```

The entries in the SCHEDxx member are normally read at OS/390 IPL. You can cause OS/390 to re-read the member without an IPL by using the SET SCH operator

command. The PPT entries must take effect before Oracle gateway and network services are started.

Details on the SCHED xx member, the PPT, and the SET SCH command can be found in the *OS/390 MVS Initialization and Tuning Reference* and in *OS/390 MVS Systems Commands*.

Step 4: Authorize the Gateway Load Library

You must APF-authorize the gateway AUTHLOAD library.

To perform this step, first select your gateway data set name qualifiers. If an Oracle8 Enterprise Edition for OS/390 server is currently installed, then these qualifiers must be different from the qualifiers used for the Oracle Enterprise for OS/390 server. The disk volume for the gateway AUTHLOAD library is also specified in this step.

In this guide, these first two qualifiers are denoted as *oran.orav*.

Because future installation procedures for updates or service might require Oracle modules to reside in their own authorized library, Oracle Corporation does not recommend installing the authorized Oracle load modules in an existing authorized library.

To mark the gateway load library as authorized, use ISPF EDIT, the IEBUPDTE utility, or an equivalent function to add the gateway load library name to the appropriate PROG xx or IEAAPF xx member in SYS1.PARMLIB (where xx is the value in the APF= xx parameter specified in IEASYS nn or by the system operator when the OS/390 system is IPLed).

You can also authorize the load library in the JCL for the gateway as you proceed through the installation. For example:

```
// SETPROG APF,ADD,DSN=name_of_your_load_library,VOL=VOLSER
```

This is generally the simplest method of APF authorizing a library.

Step 5: Re-IPL OS/390 or Use Dynamic OS/390 Commands

On systems that support dynamic subsystem definition the subsystem name can be defined dynamically by using the SETSSI system command. The Oracle AUTHLOAD library can be APF-authorized dynamically by using the SETPROG system command. You can use these commands to avoid an IPL. However, you still need to make the changes to the SYS1.PARMLIB members as discussed in ["Step 1: Define OSDI Subsystem for Oracle Gateways"](#) and ["Step 4: Authorize the](#)

[Gateway Load Library](#)". These changes to the SYS1.PARMLIB members are necessary for the subsystem and APF entries to be permanent.

You must IPL OS/390 or use the SETSSI and SETPROG system commands to ensure the subsystem is defined and the Oracle AUTHLOAD library is APF-authorized before they attempt to initialize the database.

Configuring the Gateway

This chapter describes the steps to configure the Oracle Transparent Gateway for DB2. The following topics are included:

- [Introduction](#) on page 5-2
- [Checklists](#) on page 5-2
- [Configuration Steps](#) on page 5-3
- [Post Configuration Steps](#) on page 5-36

The configuration tasks begin on page 5-2

Introduction

After defining the OSDI subsystem, and installing the gateway as described in [Chapter 4, "Installation"](#), the gateway must be configured. This process is divided into two portions:

- Configuration, which must be performed after installation has been completed
- Post Configuration, optional configuration which may be performed anytime after Configuration has been completed

To perform the configuration tasks, you might need system administrator authority on the DB2 subsystem. If so, obtain DB2 SYSADM authority before proceeding with the configuration.

The high-level and second-level qualifiers that are used in the gateway installation are oran.orav. Substitute your high-level and second-level qualifiers for oran.orav.

Checklists

Checklists are provided for Configuration and Post Configuration.

Configuration Checklist

Use the steps that follow to configure the gateway:

- Step 1: Configure a Gateway Service using the ISPF panels
- Step 2: Run the sidJB00 Configuration Job
- Step 3: Run the sidJC01 and sidJC02 Configuration Jobs
- Step 4: Make Authorization and Local Date Exits Available to DB2
- Step 5: Run the Scripts to Create Required Tables and Views in DB2
- Step 6: Bind the DB2 Package
- Step 7: Bind the DB2 Plan
- Step 8: Grant EXECUTE on DB2 Plan
- Step 9: Edit the PARMLIB Members
- Step 10: Gateway Service Definition
- Step 11: Setup Gateway JCL
- Step 12: Start the Gateway

Post Configuration Checklist

- ❑ Step 1: Move Reentrant Modules to OS/390 Link Pack Areas
- ❑ Step 2: Examine Oracle Dump Data Sets and Modify as Necessary
- ❑ Step 3: Examine Oracle Trace Data sets and Modify as Necessary

Configuration Steps

Configuration consists of twelve (12) steps.

Step 1: Configuring a Gateway Service Using ISPF Panels

The following steps guide you through the process of setting up your user environment to run the Oracle Installation Dialog Facility to configure an Oracle gateway. The Oracle software for this release must have already been installed.

Step 1.1: Selecting Oracle ISPF Profile Files

The Oracle installation and customization process uses two ISPF profile files: the `tso_userid.ORISPF.O023PROF` sequential data set and the `O023PROF` member of the `tso_userid.ISPF.ISPPROF` PDS. Both of these files are created automatically if they do not exist when you invoke the Oracle installation and customization process. However, if you are going to configure an Oracle gateway after someone else performed the product installation, you must use the profile files of the person who performed the product installation.

If you performed the Oracle product installation under the same TSO userid that you are using for the other customization, then proceed to "[Step 1.2: Execute ISPF and Invoke the Oracle Installation and Customization Process](#)".

If you did not perform the Oracle product installation under the same TSO userid that you are using for the other customization, then you must:

- Copy the `O023PROF` member from `tso_userid.ISPF.ISPPROF` of the TSO userid under which the Oracle product installation was performed to the `tso_userid.ISPF.ISPPROF` of the TSO userid under which you are performing the other customization.
- Copy or rename sequential data set `tso_userid.ORISPF.O023PROF` of the TSO userid under which the Oracle product installation was performed to a sequential data set named `tso_userid.ORISPF.O023PROF` of the TSO userid under which you are performing the other customization.

Step 1.2: Execute ISPF and Invoke the Oracle Installation and Customization Process

Invoke ORIPO01 to start the customization process.

To invoke the ORIPO01 CLIST, logon to TSO, enter ISPF, and select option 6 (TSO Command Processor) from the main ISPF menu. From option 6, use the EXEC command to invoke the ORIPO01 CLIST from the ISPCLIB data set created in Step 2:

```
----- ISPF COMMAND SHELL -----
Enter TSO or Workstation commands below:
===> EXEC 'oran.orav.ISPCLIB(ORIPO01)'
```

If you are using the correct ISPF profile files, panel OR@PRIM should be displayed immediately. Proceed to ["Step 1.3: Select Product Set to install \(Panel OR@PRIM\)"](#).

If you see prompts on the screen prior to the display of OR@PRIM (prompts that are asking for data set name qualifiers), then you did not get the correct ISPF profile files. Return to ["Step 1.1: Selecting Oracle ISPF Profile Files"](#).

Step 1.3: Select Product Set to install (Panel OR@PRIM)

Place an S in front of "Oracle Transparent Gateway for DB2", leaving the high-level and second-level qualifiers that were used during the product installation process as they are.

When you select "Oracle Transparent Gateway for DB2" on panel OR@PRIM and press [Enter], panel OR@INST is displayed. You are led through a series of panels that let you select and customize the Oracle Transparent Gateway for DB2 for your installation.

Step 1.4: Select Installation/Customization Option to be Performed (Panel OR@INST)

Different installation and customization options may be performed from this panel. These options go in pairs: a definition option which takes you through a series of ISPF panels and a generate option which produces a job stream based on the information provided on the ISPF panels to further populate the Oracle INSTLIB library. (The generate option is typically followed by other tasks to complete the installation or customization that you are performing.)

From the Oracle Primary Option Menu, select option 5 "Define New Gateway for DB2 Parameters".

Step 1.5: Begin Gateway Parameter Definition (Panel ORNEWGD)

After selecting the "Define New Gateway for DB2 Parameters" definition option from the Primary Option Menu, you will see panel ORNEWGD. This panel shows a list of the various gateway customization parameters that must be provided in order to complete the customization process.

To begin the new gateway definition process, press [Enter] to begin with option 1. The steps that follow describe the procedure and the customization panels that you encounter.

The following rules apply to the customization panels:

- Most customization parameters have default values you can change by typing in a new value in place of the default value. Other required parameters have no default values, and you must supply values for them before proceeding to the next panel.
- You can move forward from one panel to the next by entering C on the command line.
- Most panels allow you to return to the previous panel by pressing the [PF3] key.

Step 1.6: Specify Oracle Transparent Gateway for DB2 OPTIONS (Panel ORGDIP10)

1. Gateway for DB2 service ID

Enter the service identifier for the gateway that you want to create. The value that is specified here will be used as the OSDI gateway service identifier as well as the service name. The SID is limited to 4 characters on this panel because it is used to generate PDS member names in the INSTLIB and PARMLIB libraries for jobs and parameter files that are used later in the gateway customization process.

2. Character set name

You can use hexadecimal values of 37, 37C, 500, or 500C for character set names in this parameter. If you use a character set name other than one of these, then you must change it manually in the environment variables member. Refer to [Chapter C, "National Language Support"](#), for information. Refer also to [Chapter 2, "Release Information"](#).

3. Oracle RECO user ID

Specify a valid DB2 userid to be used for DB2 recovery during Oracle recovery processing. This ID requires no special privileges and should not have TSO logon authority. This value must be defined or the gateway will not come up.

4. Subsystem Name of OSDI

Enter the OSDI subsystem name under which you will run this gateway service. Note: This OSDI subsystem must be configured and initialized prior to starting the gateway service. Refer to the ["Post-Installation Steps"](#) on page 4-14 for more information.

5. Maximum Number of Sessions per Address Space

Enter the maximum number of sessions that are allowed to run in this gateway address space. Refer to ["MAX_SESSIONS | MAXSESS"](#) on page 5-19 for information about the MAX_SESSIONS gateway region parameter.

6. Initial Size of the C Stack

Enter the size of the C stack to be allocated initially for each session. Refer to ["INIT_STACK_SIZE | INTSTKSZ"](#) on page 5-19 for information about the INIT_STACK_SIZE gateway region parameter.

When information in this panel is correct, enter C to continue to the next panel.

Step 1.7: Specify DB2 Related Information (Panel ORGDIP20)

1. Target DB2 subsystem name

Specify the four character DB2 subsystem name. The default is DSN0.

2. DB2 plan name

Specify the DB2 plan name under which the DBRM for the gateway is bound. The default is G4DB2PLN.

3. Qualifier for bind plan

Specify the qualifying userid to be used to bind the plan. This userid is the owner of the plan and of the Oracle two-phase commit table. This value must be set to OTGDB2.

4. DB2 data set high level index

Specify the high-level qualifier of the DB2 load libraries. The default is DSN510.

5. Placement of new DSNEXT load library

Specify the unit and volume serial number of the new library, *oran.orav.DSNEXIT*. This library is created as part of the *sidJCO2* job. It contains the new DB2 date and logon exits.

6. Database name for gateway Data Dictionary

Specify the name of the database in which the gateway data dictionary tables and views exist. The default is *DSNDB04*.

When the information in this panel is correct, enter *C* to continue to the next panel.

Step 1.8: Specify the Oracle Transparent Gateway for DB2 Exits (Panel ORGDIP30)

G4AUTH Use *G4AUTH* for the DB2 logon exit information. For more information, refer to "[Gateway Security](#)" on page 7-4 in [Chapter 7, "Administering the Gateway"](#).

1. Write logon messages to system console?

Reply **Y** if you want debugging messages written to the system console. Debugging messages include the userid and address of critical control blocks. Messages are also written to the system console if security checking fails.

2. Allocate memory below 16M line?

Reply **Y** to force all allocations of memory below the 16M line. Certain OS/390 facilities require addresses below 16M. If you modify your logon exit to include any of these OS/390 facilities, then reply **Y**.

3. Test mode (bypass security checking)?

Reply **Y** to bypass all SAF calls. Replying **Y** to this option allows you to perform initial tests, such as the installation verification program (IVP). The module is linked under the name *TESTAUTH* if you choose this option. Reply **N** to implement standard security checking. This option can be used to test your gateway installation without security checking. After verifying successful installation, you can reset this option to implement or test security checking for your production system.

4. Check access to DB2 security profile?

Reply **Y** to check the user's access to an optional DB2 security class, *DSNR*. The entity name is *db2_subsys_name.BATCH*. Specify **N** if this class is not defined.

DSNXVDTX Use DSNXVDTX to indicate if DB2 date exit information is to be included. For more information, refer to [Chapter 9, "Developing Applications"](#).

1. Include this DB2 date exit?

Enter Y or N. The default is N. Use of this exit causes assemble and link steps to be generated at the end of the sidJC02 job.

DB2AUTH Use DB2AUTH for DB2 connection exit information. For more information, refer to ["Gateway Security"](#) on page 7-4 in [Chapter 7, "Administering the Gateway"](#).

1. Write logon messages to system console?

Reply **Y** if you want debugging messages to be written to the system console. Debugging messages include the userid and address of critical control blocks. The first three secondary userids are also written to the system console.

2. Set current SQLID with a secondary id?

Reply **Y** to allow the gateway users to access tables owned by their secondary ID without prefixing the table name with the secondary ID. If you reply **Y**, then the first in the list of secondary ids is used as the current SQLID.

If you reply **Y**, then you might have to customize the sample exit.

3. Entry point of current DSN3@ATH?

The exit for the gateway is DSN3@ATH. If another product has linked its own copy of DSN3@ATH, then specify the entry point CSECT name of the module. The default is DSN3@ATH.

When the information in this panel is correct, enter C to continue to the next panel.

Step 1.9: Specify File Processing Parameters (Panel ORGDIP35)

This panel allows you to specify the file management parameters for the default file group (DFLT). Refer to [ORA\\$FPS](#) on page 5-32 for information about these parameters.

When the information in this panel is correct, enter C to continue to the next panel.

Step 1.10: INSTLIB/ISPSLIB File Tailoring Information (Panel ORGDIP40)

1. Oracle ISPSLIB (skeleton) Library member

This item is showing the file tailoring skeleton member that will be used during the generate step. You cannot change this value.

2. Oracle Installation Library

Provide information about the INSTLIB library in which the installation jobs are built. This is normally the same data set that is created when the first file is downloaded from the Oracle distribution tapes (in "[Step 1: Load the Oracle Installation JCL](#)" on page 4-3). However, you may change the Oracle INSTLIB library name if required. Enter **NEW** or **SHR** for DATA SET DISPOSITION. The DATA SET DISPOSITION defaults to SHR. If the data set is new, then specify a VOLUME SERIAL and DEVICE TYPE.

When the information in this panel is correct, enter C to continue to the next panel.

Step 1.11: Define JOB Cards for Install Jobs (Panel ORGDIP50)

This panel allows you to define the JOB card structure that will be used with each of the generated installation jobs. The additional lines can be used to add JES subsystem control cards or similar requirements. The JOBNAME parameter and the NOTIFY parameter default to your TSO logon userid. Change the default information as needed.

When the information in this panel is correct, enter C to continue to the next panel.

Step 1.12: New Gateway Definition Complete (panel ORGDIP90)

The appearance of this panel indicates that you have completed the New Gateway Definition process. You may use [PF3] to back up through the previous panels to make any corrections that you need. Or if you are satisfied with all the parameter values in the customization panels, you may use [PF4] to return to the Primary Option Menu panel.

Step 1.13: Generate Gateway Job

Select option 6 (Generate Gateway job) from the Oracle Primary Option Menu panel. The installation CLISTs generate a jobstream in a new member, ORGDJA01, in the Oracle INSTLIB library.

The output displays as the generate step is executing. When you see the GENERATION PROCESS COMPLETED message, press [Enter] to return to the Oracle Primary Option Menu. Then press [PF3] to exit the install dialog.

Step 1.14: Run the Gateway Job (ORGDJA01)

You do not need to edit the generated job, ORGDJA01, although you might want to change the jobname. Submit this job after you have made any changes appropriate for your site.

Job ORGDJA01 creates multiple members in the INSTLIB library, including a series of configuration jobs with member names *sid*JB00 through *sid*JD00 where *sid* is the gateway service identifier that was specified on panel ORGDIP10.

Examine the output from the ORGDJA01 job to confirm its successful execution. The return code from ORGDJA01 should be 0, indicating successful execution.

Step 2: Run the *sid*JB00 Configuration Job

This job copies customized parameter members to the Oracle PARMLIB library. These members include:

<i>sid</i> ENV	Gateway environment variable file
<i>sid</i> PARM	OSDI gateway region parameter file
<i>sid</i> SVC	OSDI gateway service definition file
<i>sid</i> FPS	gateway file processing parameter file

where *sid* is the gateway service identifier specified during the gateway customization process.

These parameter files should be reviewed and modified as necessary for your installation, because some of the default values that are used may not be appropriate for your site.

The *sid*SVC PARMLIB member contains the OSDI DEFINE SERVICE and START commands for the gateway service. This member is not intended to be used as is. You should copy the contents of this member into your OSDI subsystem parameter file so that the gateway service will be defined and started every time the OSDI subsystem is initialized. Refer to "[Step 10: Gateway Service Definition](#)" on page 5-29 for more information.

The *sid*FPS PARMLIB library member contains only a Default file group entry. If the Default file group parameters are not suitable for all the file groups, you may add to this member the other file groups with the parameters that you need. Refer to "[ORA\\$FPS](#)" on page 5-32 for more information.

The *sid*ENV and *sid*PARM PARMLIB library members only contain a subset of the respective parameter values. You may add or remove as necessary from these files.

This job also copies a customized gateway procedure to the system procedure library that was specified during the product installation process. The procedure is named `G4sid` where `sid` is the gateway service ID that was specified during the gateway customization process.

This job copies customized SQL scripts to the Oracle `G4DB2.SRCLIB` library. These members include:

<code>sidGRNT</code>	Grants execute privilege n the Gateway for DB2 plan
<code>sidDDT</code>	Creates the Gateway for DB2 Data Dictionary
<code>sidDLVW</code>	Drops the Gateway for DB2 Data Dictionary

Step 3: Run the `sidJC01` and `sidJC02` Configuration Jobs

Run configuration jobs `sidJC01` and `sidJC02` in the Oracle `INSTRIB` library to continue the configuration process. The `sidJC01` configuration job assembles the gateway logon security exit and the `sidJC02` configuration job assembles the DB2 logon exit and local date exits.

Both configuration jobs reference data set `SYS1.AMODGEN`. This data set might be named `SYS1.MODGEN` at your installation. If it is, then edit the `sidJC01` and `sidJC02` configuration jobs to change the name of this data set. Both of these configuration jobs reference Assembler H (IEV90). Your installation might have High Level Assembler (ASMA90). If it does, then edit these configuration jobs to change the Assembler.

For additional information regarding gateway security, refer to "[Gateway Security](#)" on page 7-4 in [Chapter 7, "Administering the Gateway"](#).

Step 4: Make Authorization and Local Date Exits Available to DB2

Perform this step in one of two ways:

- Add the APF-authorized gateway exit library from the `sidJC02` job to the DB2 startup procedures `DBM1` and `MSTR`.
- Copy the gateway exit `DSN3@ATH` to the DB2 exit library. If a local date exit is required, then copy the `DSNXVDTX` exit. The gateway exits are in `oran.orav.DSNEXIT`.

The DB2 Local Date Exit (`DSNXVDTX`) provided with the gateway supports Oracle date formats 'DD-MON-RR' and 'DD-MON-YYYY'. This requires a DB2

installation update to set the DB2 local date length. Option 10 on DB2 install panel DSNTIPF, which specifies the local date length, must be set to 11, if not already set as 11.

You must stop and start the DB2 subsystem to access this DB2 logon, DB2 local date exits, and the DB2 local date length parameter.

Step 5: Run the Scripts to Create Required Tables and Views in DB2

For the DB2 database named on the ORGDIP20 configuration panel, use the DB2 SPUFI utility or batch job to run the following SQL scripts for all gateway installations:

- `oran.orav.G4DB2.SRCLIB(sidDDT)`
- `oran.orav.G4DB2.SRCLIB(G4DDVIEW)`

Attention: If you migrated from an lower release of Oracle Transparent Gateway for DB2, then you must first run *sidDLVW*. Refer to [Chapter 11, "Migration and Coexistence with Existing Gateways"](#), for more information.

Step 6: Bind the DB2 Package

Oracle Transparent Gateway for DB2 requires you to bind gateway DBRM G4DB2PLN to 10 separate packages. The 10-package mechanism enables a maximum of 5000 DB2 cursors per session while keeping EDMPOOL usage under control.

When you bind a package, you specify the collection to which the package belongs. These 10 packages should be included in the package list when binding the gateway plan. You can bind the packages in either of two ways:

- In a batch job, with the JCL supplied in the *sidJD00* member of the Oracle INSTLIB library. The JCL supplied in this member performs both the bind package and bind plan.
- Interactively, using DB2I.

Use the IBM DB2I Bind Package panel options to bind each package. This step must be performed 10 times, once for each collection ID (V92011A through V92011J). The panels that are discussed below are for DB2 release 5.1, release 6.1, and release 7.1. In some cases, improved performance can be achieved by selecting different options from those discussed. Consult with your DB2

administrator to determine which options best suit your particular environment.

On the DB2I Bind Package panel, use the appropriate settings for your installation. Following is a description of each option:

- Option 1 specifies which DB2 system to use to bind the package. If left blank, then this option defaults to the local DB2 system.
- Option 2 specifies the DB2 collection in which the package is located. The following naming convention must be used for the collection ID: V92011x, where x is any letter in the range A through J. The bind package must be performed 10 times, once for each collection ID from V92011A through V92011J.
- Option 3 specifies whether you are creating a new package or making a copy of an existing package. You must specify DBRM for this option.
- Option 4 specifies which DBRM member to bind. You must specify G4DB2PLN for this option.
- Option 5 specifies a password for the library name listed in the LIBRARY field. You can leave this option blank.
- Option 6 specifies the name of the library containing the DBRM specified in the MEMBER field, G4DB2PLN. In this example, the DBRM is located in data set *oran.orav.G4DB2.SRCLIB*.
- Option 7 specifies whether to change current defaults. You must specify YES to be able to change current defaults.
- Option 8 specifies whether to enable or disable other IBM intersystem connection types to use with this package. You should specify NO for this option.
- Option 9 specifies the primary authorization ID owning the package.
- Option 10 should be left blank so that the qualifier specified on the Bind Plan panel takes precedence.
- Option 11 specifies whether to replace an existing package or add a new one. You should specify REPLACE for this option.
- Option 12 specifies whether to replace a specific version of the package or create a new one. You should leave this option blank.

For additional information on the options for the Bind Package panel, refer to the IBM documents for your platform and operating system.

When you complete the changes on the Bind Package panel, press Enter to continue.

- Option 1 specifies the isolation level of the package. You should leave this option blank so that the isolation level specified on the Bind Plan panel takes precedence.
- Option 2 specifies when to validate DB2 objects or privileges for the package. You should specify BIND for this option.
- Option 3 specifies when to release locks on resources. You should leave this blank so that the resource release time specified on the Bind Plan panel takes precedence.
- Option 4 specifies whether to obtain EXPLAIN information on how SQL statements in the package run. You can leave this option blank.
- Option 5 specifies whether data currency is required for ambiguous cursors when the isolation level of cursor stability (CS) is in effect. This option also determines whether block fetching can be used for distributed, ambiguous cursors.

Although cursors used in block fetch operations result in reduced network traffic, you are still vulnerable to reading data that has already changed. In a block fetch, DB2 fetches as many rows as can fit into a buffer before sending the entire buffer over the network. During that time, the underlying data might have been modified before the application actually asks for the data. Situations such as fetching a row of values that no longer exists, or missing a recently inserted row, can occur. If these types of situations are acceptable, then data currency is not required (CURRENTDATA=NO). If data currency is required (CURRENTDATA=YES), then be aware that a separate buffer is sent over the network for each row fetched.

- Option 6 specifies the use of parallel processing (if possible) to run queries. You can leave this option blank.
- Option 7 specifies whether runtime rules or bind time rules apply to dynamic SQL statements at run time. You should specify RUN for this option.
- Option 8 should be left blank.
- Option 9 specifies whether to have DB2 determine an access path at runtime using values for host variables, parameter markers, and special registers. You should set this option to NO.
- Option 10 specifies whether to defer preparation for dynamic SQL statements that refer to remote objects. You should set this option to NO.

- Option 11 determines whether DB2 keeps dynamic SQL statements after commit points. You should set this option to NO.

For additional information on the options for the Defaults for Bind Package panel, refer to the IBM documents for your platform and operating system.

When you complete the changes on the Defaults for Bind Package panel, press Enter to continue. Repeat this step, varying only the collection ID, until all packages (V92011A through V92011J) are bound.

Step 7: Bind the DB2 Plan

You can bind the gateway plan in either of two ways:

- As a batch job

Use a batch job with the JCL supplied in the *sidJD00* member of the Oracle INSTLIB library. The JCL supplied in this member performs both the bind package and the bind plan.

Note: If the *sidJD00* job has run successfully in "Step 6: Bind the DB2 Package", then it does not need to be run again.

You can proceed to "Step 8: Grant EXECUTE on DB2 Plan".

- Interactively, using DB2I

Use the IBM DB2I Bind Plan panel options to bind the plan. (The panels discussed are for DB2 release 5.1.) In some cases, improved performance can be achieved by selecting different options from those discussed in this section. Consult with your DB2 administrator to determine which options best suit your particular environment.

On the IBM DB2I Bind Plan panel, use the appropriate settings for your installation. The options are:

- Option 1 specifies the first DBRM to include in the gateway plan. Specify G4DB2IX, G4DB2PRC, G4DB2PIN, and G4DB2V51 for this option.
- Option 2 specifies the password for the libraries listed in the LIBRARY field. You can leave this option blank.
- Option 3 specifies the name of the PDS containing the gateway DBRMs. In this example, the DBRMs are located in data set oran.orav.G4DB2.SRCLIB.

- Option 4 must be set to YES to specify additional DBRM members.
- Option 5 specifies the name of the DB2 application plan to create. Specify the DB2 plan name entered on the gateway configuration panel ORGDIP20. The default plan name is G4DB2PLN.
- Option 6 must be set to YES to change current defaults.
- Option 7 specifies whether to enable or disable other IBM intersystem connection types to use with this package. Specify NO for this option.
- Option 8 specifies whether to include a package list for the plan. You must specify YES for this option.
- Option 9 designates the primary authorization ID owning the plan.
- Option 10 specifies the userid from the gateway installation panel (ORGDIP20) that is used as the qualifier. This value must be set to OTGDB2.
- Option 11 specifies the size (in bytes) of the authorization cache. For details on this option, refer to the IBM documents for your platform and operating system.
- Option 12 specifies whether this plan is new or is being replaced. Specify REPLACE for this option.
- Option 13 determines whether users with the authority to bind or run the existing plan are to keep that authority over the changed plan. Specify YES for this option.
- Option 14 specifies the initial server to receive and process SQL statements. You can leave this option blank.

For additional information on the options for the Bind Plan panel, refer to the IBM documents for your platform and operating system.

When you complete the changes on the Bind Plan panel, press Enter to continue.

If you plan to use DB2 release 6.1 or later, then you need to add a third DBRM (G4DB2V61) to the IBM DB2I Bind Plan (Part 2 release 6.1 or 7.1) panel.

Confirm the DBRM library. Press the PF3 key to continue.

The IBM DB2I Defaults panel lets you set the ISOLATION LEVEL and RESOURCE RELEASE TIME to those values applicable to your site. ISOLATION LEVEL specifies the ways in which read-only operations are isolated from the effects of concurrent write operations.

The gateway does not change the locking behavior of the DB2 database. Therefore, it is important to understand how the `ISOLATION LEVEL` can impact the behavior of Oracle applications accessing DB2 through the gateway.

Oracle applications written for the Oracle database server and deployed later to access DB2 through the gateway might require special consideration. Oracle read operations do not prevent concurrent write operations and, therefore, do not have to `COMMIT` in order to allow subsequent updates to the Oracle database server. Thus, many Oracle applications do not `COMMIT` at the end of read operations.

DB2 generally requires a `COMMIT` to release read transaction locks to allow subsequent write operations by other DB2 transactions. If a previously developed Oracle application that does not run a `COMMIT` statement after reading data is redirected to a DB2 database, then DB2 might prevent subsequent DB2 transactions from updating the data. This is because, unlike the Oracle database server, DB2 acquires locks when data is read.

Refer to the IBM documents for your platform and operating system before setting the `ISOLATION LEVEL` to one of the following settings:

- Cursor stability (CS) can be chosen to work with the gateway for maximum concurrency with data integrity. However, if a read-only application does not `COMMIT` or close the cursor, then the CS isolation level might result in read locks being set that would prevent concurrent and subsequent write operations.
- Uncommitted read (UR). This isolation level can be used with the gateway to allow Oracle applications that do not `COMMIT` with read transactions to access DB2 without preventing write operations. It is important to note that the UR isolation level allows gateway applications to read uncommitted data.
- Repeatable read (RR) results in locks being held on all rows or pages accessed, preventing concurrent and subsequent write operations until the application performs a `COMMIT`.

Ensure the other parameters are specified as listed.

On the IBM DB2I Package List for Bind Plan panel, fill in the collection names and package ids to include in the gateway plan. If you plan to run DB2 stored procedures through the gateway, then insert an asterisk (*) for the collection ID and package ID as the last entry in the `COLLECTION` and `PACKAGE-ID` columns.

When you complete the changes on the Package List for Bind Plan panel, press the PF3 key to bind the plan.

Step 8: Grant EXECUTE on DB2 Plan

Using the DB2 SPUFI utility, grant EXECUTE authority to public on the DB2 plan by running the *sidGRNT* member in the gateway INSTLIB library.

Step 9: Edit the PARMLIB Members

Edit the following members of the PARMLIB library to ensure the gateway parameters are set appropriately for your installation:

***sid*PARM (Gateway Region Parameters)**

OSDI gateway region parameters are supplied in a data set whose name is specified as the PARM string in the service definition. This will typically be a member of the PARMLIB PDS. Because the data set name is supplied via the gateway service PARM mechanism, no DD statement is coded in the region JCL. The data set is dynamically allocated, opened, and read when the service is started. Changing parameters in the data set has no effect until the service is stopped and restarted.

The OSDI gateway region parameters consist of a parameter name followed by the parameter value in parentheses. Each parameter has a long descriptive name and a shorter name of eight characters or less. Each record may contain only one parameter. No continuation is allowed. Records beginning with an asterisk (*) are treated as comments and are ignored. Embedded spaces and all characters after the closing parenthesis are ignored.

DEDICATED_TCB | DEDTCB

DB2 requires that each connection run under its own TCB. For this reason, the gateway must associate each user session to a dedicated TCB. This is accomplished with the OSDI DEDICATED_TCB parm:

```
DEDICATED_TCB ( Yes | No | Auto )
```

This parameter value must be set to AUTO to prevent DB2 errors.

SERVER_LOADMOD | SRVRLMOD

SERVER_LOADMOD specifies the name of the service load module, as follows:

```
SERVER_LOADMOD ( loadmod )
```

where *loadmod* is the name of the load module to load. For the Oracle Transparent Gateway for DB2, this value must be set as G4DB2SRV. This parameter is required.

LOGON_AUTH | LGNAUTH

LOGON_AUTH specifies the action that will occur at each connection through a dblink. The syntax is:

```
LOGON_AUTH ( exitname )
```

where *exitname* is the 1- to 8- character name of the gateway logon security exit load module created in Step 3 of the Gateway configuration process. The exit module must reside in the system linklist or in an APF-authorized library that is part of the gateway region STEPLIB concatenation.

Example:

```
LOGON_AUTH (G4AUTH)
```

For more information about gateway logon security exit, refer to "Gateway Security" on page 7-4 in Chapter 7, "Administering the Gateway".

INIT_SESSIONS | INITSESS

The INIT_SESSIONS parameter specifies how many sessions to start in the gateway address space, as follows:

```
INIT_SESSIONS ( number_of_sessions )
```

where *number_of_sessions* is the number of sessions to start in an address space. The default is zero. The default of zero should normally be used.

INIT_STACK_SIZE | INTSTKSZ

INIT_STACK_SIZE controls the size of the C stack that is allocated for each session, as follows:

```
INIT_STACK_SIZE ( init_size )
```

where *init_size* determines the initial size of the C stack. This value can be specified as *n* or *nk*. The default is 128K. For more information on INIT_STACK_SIZE, refer to "The User Stack Area in OS/390" on page 16-10 of the *Oracle9i Enterprise Edition System Administration Guide for OS/390*.

MAX_SESSIONS | MAXSESS

The MAX_SESSIONS parameter limits the number of sessions that can be scheduled in an address space, as follows:

```
MAX_SESSIONS ( number_of_sessions )
```

where *number_of_sessions* is the maximum number of sessions per gateway address space. This value can be specified as *n* or *nK*. The default is 1024. The number of sessions that can be supported in an address space depends on the complexity of the work. Limiting the number of sessions per address space reduces the chances of session failure due to exhaustion of virtual storage.

SMF_STAT_RECNO | SMFSTRCN

SMF_STAT_RECNO specifies the SMF record number to use, as follows:

```
SMF_STAT_RECNO ( record_number )
```

where *record_number* is the number of the desired record of Oracle SMF statistics. The default is zero (0). Otherwise, the value must be specified between 128 and 255 for this parameter. Example:

```
SMF_STAT_RECNO (199)
```

The collection and writing of Oracle SMF statistics records is controlled by this single parameter in the OSDI service parameter file. A zero (0) for this parameter indicates that no SMF statistics record is to be written. The SMF record number that is chosen must not be the same as the number that is used by any other OS/390 software. This includes any OS/390 Oracle server instances and MPM gateway instances. Due to the differences in the report formatting programs, OSDI gateways and MPM gateways should use different SMF record numbers.

If this parameter is not specified, or if zero is specified, then no SMF statistics collection or recording is done. This saves some CPU overhead and saves the overhead of the SMF write itself (which is mostly asynchronous work done by the SMF address space, the in-line overhead is mainly just moving data into SMF buffers). For more information about SMF, refer to Appendix D, "The Oracle SMF Interface".

If there is an active OSDI database instance, you should specify a different SMF record number for this gateway instance.

TRACE_DSNAME | TDSN

TRACE_DSNAME specifies the destination for Oracle gateway trace files. This includes traces requested by setting gateway trace TRACELEVEL in the environment member (ORA\$ENV).

```
TRACE_DSNAME ( filespec )
```

where *filespec* is either a SYSOUT specification (including class, form, and JES destination) or a data set name.

A SYSOUT specification is of the form:

```
//SYSOUT: c, form, dest
```

where *c* is a SYSOUT class, *form* is a forms identifier, and *dest* specifies a destination. All three positional parameters are optional and default to whatever defaults are established for the gateway address space. Note that the default class is equivalent to specifying SYSOUT=*. The leading slashes can be omitted, and the word SYSOUT can be abbreviated to just the letter S.

The following are all examples of valid SYSOUT file specifications:

```
S:
//SYSOUT:*
//S:X,,NPFPRINT
SYSOUT:* ,STD
```

As an alternative to a SYSOUT specification, you can specify a data set name. Because each trace file created as a data set must have a unique data set name, the supplied value must include system symbols that guarantee uniqueness. Refer to Appendix C of the *Oracle9i Enterprise Edition System Administration Guide for OS/390* for more information.

To guarantee uniqueness, use some combination of the session identifier (&ORASESST) system symbol, date (&LYMMDD), and time (&LHHMMSS). Also use high-level qualifier(s) that are appropriate for your installation. This will avoid the possibility of duplicating trace data set names generated in other Oracle instances you run. All components of the string must resolve to produce a name that is valid for an OS/390 sequential data set. For example:

```
TRACE_DSNAME(ORA3A.TRACE.D&LYMMDD..T&LHHMMSS..&ORASESST)
```

The allocation parameters for Oracle trace data sets are obtained from the DBTR file entry in the file management parameters, discussed on page 5-23 in the "ORA\$FPS:" section. If this parameter is omitted or fails to produce a valid, unique data set name, all Oracle trace files are written to the default SYSOUT class associated with the server region.

***sid*ENV (Environment variables and HS Initialization parameters)**

There are two types of parameters for member *sid*ENV of the PARMLIB library:

- The first type is parameters that set an environment variable. An environment variable is used to direct internal gateway processing.

- The second type is heterogeneous services (HS) initialization parameters that are uploaded to the Oracle database server during the first connection of a session.

Parameters for Environment Variables in Member *sidENV*

Following are the valid parameters and their defaults for the environment variables in member *sidENV* of the PARMLIB library. Set these parameters to values that are applicable for your site:

Table 5–1 Valid Parameters for Environment Variables of Member *sidENV*

Parameters	Default
CURRDEGREE	1
DB2DESCTAB	YES
DB2LONGMSG	NO
DB2WARNING	NO
DB2STATS	NO
DB2READONLY	NO
FDS_CLASS_VERSION	9.2.0.1.1
FLUSH_CACHE_ON_COMMIT	NO
TARGET	DB2
DB2SSN	DSN0
DB2PLAN	G4DB2PLN
ORARECID	no default
NLS_LANG	no default
NLS_DATE_FORMAT	no default
NLS_NCHAR	no default
ORA_MAX_DATE	no default
TRACELEVEL	0
TO_NUMBER_OFF	NO
CNV_LIT_FMT	NO
EMPTYSTR_TO_NULL_OFF	NO

Table 5–1 Valid Parameters for Environment Variables of Member sidENV

Parameters	Default
SELECT_CONCAT_ON	NO

CURRDEGREE

The CURRDEGREE parameter allows you to specify whether parallel query operations are to be allowed on your DB2 system. The valid values are:

- 1 turns parallel query operations off. The default value is 1.
- ANY specifies parallel query operations are to be allowed on your DB2 system. Parallel processing must be enabled on your DB2 system before parallel query operations can occur.

For further information on DB2 parallel I/O and CP processing, refer to the IBM documents for your platform and operating system.

DB2DESCTAB

The DB2DESCTAB parameter specifies whether the gateway uses a DB2 DESCRIBE TABLE for acquiring DB2 table information. The default of DB2DESCTAB set to YES directs the gateway to run a DESCRIBE TABLE when acquiring information about the DB2 table. This can provide performance enhancements, but cannot be used when accessing DB2 tables using a DB2 alias. DB2 does not allow a DESCRIBE TABLE against a DB2 alias. DB2DESCTAB set to NO prevents the DESCRIBE TABLE from being used by the gateway so applications can access DB2 tables using the DB2 alias.

DB2LONGMSG

The DB2LONGMSG parameter allows longer error messages to be returned by the gateway. The default of DB2LONGMSG set to NO returns some of the error messages from DB2, but does not always return the entire error message. By setting DB2LONGMSG to YES, you ensure the entire DB2 error message can be returned by the gateway.

DB2WARNING

The DB2WARNING parameter enables you to configure the gateway so DB2 warning messages are not returned to the application as errors.

Warning messages are displayed by DB2 when SELECT statements are not formatted properly, but some default action by DB2 can still be taken to complete the task. Setting DB2WARNING to YES allows the warning message to be returned by the gateway.

The warning message is interpreted as an error by the Oracle application. The default setting of DB2WARNING set to NO processes the SELECT statement without returning the warning to the application. The gateway continues DB2 default completion without notification to the application that a warning occurred.

DB2STATS

DB2STATS specifies whether or not the gateway is to pass DB2 statistics to the Oracle Optimizer for improved query performance. If set to the default value of NO, then the gateway does not pass statistics. If set to YES, then the statistics are used by the Oracle Optimizer to choose the access plan for SQL statements that involve DB2 objects. This results in DB2 tables appearing to the Oracle Optimizer as if they were Oracle analyzed objects. If set to YES, then ensure the DB2 utility RUNSTATS is run against the DB2 tables accessed by the gateway. This ensures information about these tables is available in the DB2 catalog. Also ensure the gateway plan qualifier has SELECT privileges for the SYSIBM.SYSKEYS, SYSIBM.SYSINDEXES, and SYSIBM.SYSTABLES DB2 catalog tables.

DB2READONLY

DB2READONLY controls whether the gateway is enabled in read-only mode. If it is set to the default value of NO, then the gateway is not in read-only mode. If it is set to YES, then read-only capabilities are enabled and only queries are allowed through the gateway to DB2. Any SQL statements or calls that attempt to modify a DB2 object are rejected. When the gateway is in read-only mode, INSERT, UPDATE, DELETE, DB2 stored procedures, or passthrough SQL are not allowed.

For additional information, refer to [Chapter 8, "Using the Gateway"](#).

FDS_CLASS_VERSION

This parameter controls AUTOREGISTER, the uploading of class capabilities from the gateway to the Heterogeneous Services layer of the Oracle Integrating Server. When the value of this parameter changes, it causes HS to upload and use the new G4DB2CAP capabilities table.

You should rely on the default and only specify this parameter at the request of Oracle Support Services.

FLUSH_CACHE_ON_COMMIT

FLUSH_CACHE_ON_COMMIT specifies when the describe table cache is flushed. If the value is set to YES, then the describe table cache is flushed each time a transaction is committed. If the value is NO, then the describe cache is flushed when the Oracle session terminates. The default setting of NO reduces the overhead associated with flushing cached information that is retrieved repeatedly after each committed transaction. Performance might be improved by using the default setting of NO.

The following parameters are set based on values specified during the installation and configuration process.

DB2CAP

With the introduction of the dynamic capability table, the DB2CAP environment parameter is obsolete.

TARGET

The TARGET parameter is maintained for backward compatibility with Version 4 of the gateway. If you're not a previous customer of TG4DB2 v402110 using GTW_SQL.GTWPASS procedures, you do not need to specify this parm.

DB2SSN

This parameter specifies the DB2 subsystem name to be accessed by Oracle Transparent Gateway for DB2. The default parameter is DSN0.

DB2PLAN

Specifies the DB2 plan name that Oracle Transparent Gateway for DB2 uses to access the DB2 system. The default is G4DB2PLN.

ORARECID

This parameter specifies the user name that Oracle Transparent Gateway for DB2 uses when logging on to DB2 to perform recovery. This parameter has no default but must be defined for the gateway to successfully initialize.

NLS_LANG

This parameter must match the character set of the DB2 subsystem. Refer to [Appendix C, "National Language Support"](#), for detailed information.

NLS_DATE_FORMAT

Oracle Corporation recommends that this parameter be specified as 'YYYY-MM-DD'. It is used to convert DB2 dates to and from the Oracle date format.

NLS_NCHAR

This parameter specifies the character set used by the gateway for national character sets. For non-DBCS users, this parameter should not be specified. For DBCS users, it must be specified.

ORA_MAX_DATE

This parameter must be specified as 'YYYY-MM-DD'. The Oracle database server allows a maximum date value of 4712-12-31, while DB2 allows dates up to 9999-12-31. This ORA_MAX_DATE parameter has no default. If this parameter is specified, then any DB2 date value that is being sent back to the Oracle database server is inspected. If the inspected value exceeds 4712-12-31, then it is replaced by the value of ORA_MAX_DATE.

If no date value is specified for ORA_MAX_DATE, then a returned date value might not be valid to the Oracle database server. The value of ORA_MAX_DATE can be set to a value less than 4712-12-31, but the ORA_MAX_DATE is returned only if the DB2 date value exceeds 4712-12-31.

If you have no dates exceeding the year 4712, then you do not need to be concerned with this parameter.

Some examples are listed in the following table:

ORA_MAX_DATE	DB2 Value	Returned to Oracle
(not specified)	9999-12-31	9999-12-31 (invalid)
(not specified)	4713-10-24	4713-11-28 (invalid)
(not specified)	4500-11-19	4500-11-19
4712-12-31	9999-12-31	4712-12-31
4712-12-31	4713-10-24	4712-12-31
4712-12-31	4500-11-19	4500-11-19
3388-12-31	9999-12-31	3388-12-31
3388-12-31	4713-10-24	3388-12-31

ORA_MAX_DATE	DB2 Value	Returned to Oracle
3388-12-31	4500-11-19	4500-11-19

TRACELEVEL

If this parameter is set to 4, then the following information is written to the trace file:

- SQL text sent to the gateway by the Oracle database server
- SQL text sent to the target database

Set this parameter to 255 for all gateway trace information.

TO_NUMBER_OFF = YES | NO (default)

The fix for bug 2095461, "SQLCODE = -418 is returned using TO_NUMBER against a char bind variable", introduces a new initialization parm. By default, a SQL statement sent through the gateway to DB2 that includes the TO_NUMBER function with one argument, will be translated to the equivalent DB2 function, DECIMAL(). This can cause problems when the only argument is a bind variable. You can alleviate this problem by setting the new init parm, TO_NUMBER_OFF to YES in the gateway ENV member.

This will "turn off" the capability, and the TO_NUMBER function will be processed by the Oracle instance prior to sending the SQL statement to the gateway.

CNV_LIT_FMT = YES | NO (default)

Oracle and the transparent gateway would perform implicit conversion when necessary before sending SQL to DB2. A specific case would be if there is a WHERE clause such as:

```
WHERE number_variable = 'literal_string'
```

where the literal string contains comma ',' as the decimal indicator. In that case, you would need to set CNV_LIT_FMT to YES.

Related bug - 1934416

EMPTYSTR_TO_NULL_OFF = YES | NO (default)

When it's set 'YES', HS will NOT convert empty string to NULL before sending to DB2.

Related bug - 2249392

SELECT_CONCAT_ON = YES | NO (default)

By default, the CONCAT function is post-processed. When this is set to "YES", the CONCAT function is passed on to DB2 which may cause a SQLCODE -418 due to a DB2 restriction (bug 1269591).

HS Initialization Parameters for Member *sidENV*

Following are the valid HS initialization parameters that can be specified in member *sidENV*:

Table 5–2 Valid HS Initialization Parameters for Member *sidENV*

Parameters	Generic Default
HS_DB_DOMAIN	WORLD
HS_DB_INTERNAL_NAME	DB2920
HS_DB_NAME	There is no default. Normally you would specify the gateway service SID.
HS_DESCRIBE_CACHE_HWM	100
HS-NLS_DATE_LANGUAGE	value is determined by the NLS_LANG parameter
HS_OPEN_CURSORS	50
HS_RPC_FETCH_REBLOCKING	ON
HS_RPC_FETCH_SIZE	4000 (Oracle Corporation recommends this value be set to 40 000)

If GLOBAL_NAMES is set to true in the Oracle9i INIT.ORA file, then the HS_DB_DOMAIN parameter must match the DB_DOMAIN parameter in the Oracle9i INIT.ORA file.

Refer to Oracle9i Server Distributed Database Systems for additional information about the HS initialization parameters for member *sidENV*.

The following information varies from the information presented in *Oracle9i Server Distributed Database Systems*. The information presented in this guide facilitates your ability to use the Oracle Transparent Gateway for DB2:

1. HS_DB_INTERNAL_NAME parameter default can be overridden. The default is DB2920.
2. There is no default for HS_DB_NAME. You could normally specify the gateway instance SID as HS_DB_NAME. When GLOBAL_NAMES is set to

true in the Oracle integrating server, this parameter must match the name of the database link.

3. HS_LANGUAGE parameter is derived from the NLS_LANG environment variable and cannot be overridden.
4. HS-NLS_DATE_FORMAT is derived from the NLS_DATE_FORMAT environment variable and cannot be overridden.
5. The HS-NLS_NCHAR parameter is derived from the NLS_NCHAR environment variable and cannot be overridden.
6. HS_RPC_FETCH_SIZE parameter defaults can be overridden. If the HS_RPC_FETCH_REBLOCKING parameter is set to ON (the default), then the array size for SELECT statements is determined by the HS_RPC_FETCH_SIZE parameter value. The recommended value for Oracle Transparent Gateway for DB2 is 40 000. The value shipped with Oracle Transparent Gateway for DB2 is 40 000.

The HS_RPC_FETCH_SIZE parameter defines the number of bytes sent with each fetch between the gateway and the Oracle database server.

Notes: This feature can provide significant performance enhancements, depending on your application design, installation type, and workload.

The HS_RPC_FETCH_SIZE value impacts the performance for elapsed time.

Generally, a higher HS_RPC_FETCH_SIZE value can correlate with improved performance for elapsed time.

However, it is important to evaluate the requirements for elapsed time since a higher value associated with this parameter might also impact memory utilization.

Step 10: Gateway Service Definition

To create an Oracle gateway instance under OSDI, you must first define the instance as a service using the OSDI DEFINE SERVICE command. The OSDI DEFINE SERVICE command is described completely in [Appendix B, "OSDI Subsystem Command Reference"](#). Here, we cover DEFINE parameter considerations that are specific to an Oracle gateway service.

Service Name

The service name for a gateway can be anything that you want within the content limitations described in [Appendix B, "OSDI Subsystem Command Reference"](#). By default, OSDI will use the service name as the SID for the service. The SID can be specified separately, however, and is not required to be the same as the service name.

Note: If you specify a service name that is the same as any existing subsystem name in your system (Oracle database or otherwise), then you must also specify a JOBNAME parameter that is not the same as any existing subsystem. If you do not use unique names, then OSDI starts the service using the service name as the job identifier. When OS/390 processes a start for an address space whose job name or job identifier matches a known subsystem, the job runs under control of the master subsystem instead of under control of JES.

Caution: Running OSDI services under the master subsystem is not supported. This situation must be avoided by making sure that the service runs with a job name or a job identifier that is not the same as any subsystem name.

Although you can issue the OSDI DEFINE SERVICE command via an OS/390 system console or similar facility, you should put definition commands for services that you use regularly into the OSDI subsystem parameter file, after the DEFINE SERVICEGROUP command. This ensures that the service is always defined correctly and automatically when the subsystem is initialized (normally at system IPL). In our sample gateway DEFINE SERVICE command below, the command prefix has been omitted and continuation hyphens have been included as though the command were in the subsystem parameter file:

```
DEFINE SERVICE ORAGW01 TYPE(GTW) PROC(ORAGW01) -
DESC('Oracle Test Gateway Instance') -
  PARM('ORAN.ORAV.PARMLIB(sidPARM)') -
SID(GW01)
```

TYPE

The TYPE parameter for a gateway service must be specified as GTW.

PROC

This parameter specifies the name of a service JCL procedure that you will place in one of your system procedure libraries. The procedure need not exist when DEFINE SERVICE is issued, but it must be in place before the service is started. The procedure name can be anything that you choose or that the naming standards of your installation require. The requirements for this procedure are discussed in section "Step 11: Setup Gateway JCL" on page 5-22.

PARM

The PARM for a gateway service specifies the name of an OS/390 data set containing service initialization parameters. These are gateway specific parameters and are described in "*sid*PARM (Gateway Region Parameters)" on page 5-10. Typically, PARM will specify a member of a PDS (Partitioned Data Set) that is used for various Oracle parameter files. If no member name is included in the PARM string, then the specified data set must be sequential (DSORG=PS).

JOBNAME

You can also use JOBNAME to cause the service to run with a jobname different from the service name (which is used by default).

As discussed in the "Note" on page 5-21, you must specify a JOBNAME parameter if the service name matches any existing subsystem name in your OS/390 system.

SID

The SID parameter specifies a unique identifier for the service. It is a critical element in the process that is used by Oracle gateway applications to specify the gateway instance to which they need to connect. Inbound network clients specify a SID in the Oracle dblink network address string that must match the SID that is specified in DEFINE SERVICE.

Step 11: Setup Gateway JCL

The startup procedure is in the Oracle PROCLIB library in member *G4sid*, where *sid* is the gateway SID specified in the ORGDIP10 configuration panel. If necessary, rename the procedure for your installation.

This procedure must be in place before you try to start the service, and must have security access to datasets in the procedure. In particular, it must have update and create authority for the trace datasets specified by the OSDI gateway region

TRACE_DSNAME parameter. The procedure must invoke the OSDI gateway region program with an EXEC statement such as the following:

```
// EXEC PGM=ORARASC,REGION=0M
```

REGION=0M is specified to ensure that the gateway can allocate as much private virtual memory as it needs. Some OS/390 systems may prohibit or alter a REGION parameter such as this, so you might want to check with your systems programmer to make sure that the system will accept your REGION parameter.

Note that no other EXEC statement parameters are needed. The PARM parameter of EXEC is not used by the gateway region program.

In addition to the EXEC statement, the procedure will need several DD statements, as follows:

ORA\$ENV

This DD statement is required. It specifies a sequential file or PDS member containing environment variable assignment statements. Environment variables are described in "Step 7: Edit the PARMLIB Members", subsection "[sidENV \(Environment variables and HS Initialization parameters\)](#)" on page 5-21.

ORA\$FPS

This DD statement specifies a sequential file or PDS member containing OS/390-specific parameters that control data set processing in the gateway. If you omit it, then gateway file creation operations may fail unless your installation has DF/SMS ACS routines that supply defaults for data set creation parameters. The data specified by ORA\$FPS is read only at gateway service startup.

Note: When this DD statement is omitted, an IEC130I message may appear in the system log during service address space initialization. This is normal.

The ORA\$FPS parameter file contains file group definitions which are specified using `keyword(value)` syntax. Each definition must start with the keyword `FILE_GROUP(name)` and continues until the next `FILE_GROUP` keyword is encountered. Comments must start with an asterisk (*) and can begin in any column as long as comments (that are on the same line as keywords) are separated from the last keyword by at least one blank.

Keywords can be coded one per line or strung together on the same line separated by at least one blank, but a keyword (value) pair cannot be split across two lines. No defaults are defined for the parameters. The default file group (DFLT) supplies parameters for any file group that is completely omitted from the file management parameters.

With this release, the only File Groups supported by or applicable to the gateway are DBTR indicating the attribute for gateway trace files, and NTTR (network trace).

DATACLAS(classname) Specifies an SMS data class name to be specified on DEFINE CLUSTER or dynamic allocation requests to create new data sets. DATACLAS can be abbreviated DATACL.

DEFAULT_SPACE(primary secondary) Specifies default primary and secondary space quantities for a data set that is being created. The secondary quantity is optional and is ignored at this time. Both values must be numbers and are expressed in kilobyte (1024-byte) units. DEFAULT_SPACE can be abbreviated SPA.

FILE_GROUP(name) Specifies the file group to which the file management parameters belong, where name is one of the allowed 4-letter file group names. This ends any in-progress file group definition and begins a new one. FILE_GROUP can be abbreviated FILE.

MGMTCLAS(classname) Specifies an SMS management class name to be specified on DEFINE CLUSTER or dynamic allocation requests to create new data sets. MGMTCLAS can be abbreviated MGMTCL.

STORCLAS(classname) Specifies an SMS storage class name to be specified on DEFINE CLUSTER or dynamic allocation requests to create new data sets. STORCLAS can be abbreviated STORCL.

UNIT(unitname) Specifies an allocation unit name to use in dynamic allocation requests that create new non-VSAM data sets.

VOLUMES(volser) Specifies a volume serial number to use in IDCAMS DEFINE CLUSTER commands for VSAM data sets or in dynamic allocation requests that create non-VSAM data sets.

Only a single volume serial can be specified. Because of this limitation, it is recommended that you use storage management class parameters instead of explicit volumes. VOLUMES can be abbreviated VOL.

Example

Storage management parameter example:

```
* Oracle gateway file management parameters
* Trace data files
FILE_GROUP(DBTR)
DEFAULT_SPACE(200 50)          * a comment
* Default for groups not specified
FILE_GROUP(DFLT)
DEFAULT_SPACE(10000 5000)
UNIT(SYSDA) VOL(TEMP01)
```

ORA\$LIB:

This DD statement specifies a non-authorized load library from which non-executable (data) modules are fetched. The modules contain NLS data objects and messages that are associated with Oracle NLS internationalization features. Normally these modules are installed in the Oracle MMSG data set, for example ORAN.ORAV.MMSG. The ORA\$LIB DD statement is optional: if you omit it, then the Oracle server attempts to fetch messages and NLS data objects modules from STEPLIB. Do not concatenate a non-APF-authorized MMSG data set to STEPLIB in lieu of specifying ORA\$LIB.

Note: When this DD statement is omitted, an IEC130I message may appear in the system log during service address space initialization. This is normal.

SQLNET:

This DD statement specifies an input file containing Oracle Net parameters. It is required if the gateway uses any of the following:

- network data encryption
- network activity tracing
- altering of default Oracle Net file names
- outbound database links whose Oracle Net addressing requires access to an Oracle database Names server

SQLNETLG:

This DD statement is optional and is the default destination for network error messages. SYSOUT or a sequential disk data set can be specified.

STEPLIB:

This DD statement must specify the APF-authorized Oracle AUTHLOAD library that was populated during installation. The IBM LE/370 runtime library must be concatenated to it unless your installation has put LE/370 runtime into the system linklist. A typical name for the LE/370 runtime library is SYS1.SCEERUN, but it may have a different name in your system.

SYSPRINT:

This DD statement is optional. When used, the gateway alert log is written to it. The alert log is a sequential text file containing messages related to the operation of the gateway.

You can specify a disk data set or a spool file (SYSOUT) for the alert log. If you omit the SYSPRINT DD, then OSDI dynamically allocates a spool file (using SYSOUT=*) for the alert log during service startup and writes a message to the system log identifying the system-generated DDname for the allocation. If you specify a disk data set for SYSPRINT, and if an error occurs while it is being written (including an out-of-space condition), then OSDI closes SYSPRINT, dynamically allocates a spool file, and begins writing to it.

ORA@ssn:

Where ssn is the OSDI subsystem name of the OSDI subsystem. This DUMMY DD statement is necessary if you use the G4AUTH security exit without modification. The security exit uses this DD statement to determine what subsystem it's running under and whether the subsystem is active. If G4AUTH is invoked and doesn't find this DD statement, the logon will fail.

Sample Gateway Region JCL Procedure

The following is an example of a JCL procedure for a database region:

```
//G4GW01 PROC
//ORACLE EXEC PGM=ORARASC,REGION=0M
//STEPLIB DD DISP=SHR,DSN=ORAN.ORAV.AUTHLOAD
// DD DISP=SHR,DSN=SYS1.SCEERUN
// DD DISP=SHR,DSN=DSN=DSN610.SDSNLOAD
//ORA$LIB DD DISP=SHR,DSN=ORAN.ORAV.MESG
```

```
//ORA$FPS DD DISP=SHR,DSN=ORAN.ORAV.PARMLIB(GW01FPS)
//ORA$ENV DD DISP=SHR,DSN=ORAN.ORAV.PARMLIB(GW01ENV)
//ORA@G4XX DD DUMMY
//SQLNETLG DD SYSOUT=*
```

Step 12: Start the Gateway

You can start the gateway service with OSDI Start Command. For example, from SDSF, issue:

```
/G4XX START GW01
```

where G4XX is the OSDI subsystem name and GW01 is the service name.

Post Configuration Steps

The following optional steps can be performed any time after the gateway is configured.

[Step 1: Move Reentrant Modules to OS/390 Link Pack Areas](#)

[Step 2: Examine Oracle Dump Data Sets and Modify as Necessary](#)

[Step 3: Examine Oracle Trace Data Sets and Modify as Necessary](#)

Step 1: Move Reentrant Modules to OS/390 Link Pack Areas

The Oracle CMDLOAD modules that have RMODE set to ANY and are reentrant can be placed in the OS/390 extended pageable link pack area (EPLPA) to decrease storage requirements. Other modules that are linked with RMODE set to 24 and are reentrant can be placed in the OS/390 pageable link pack area (PLPA) below the 16M line. For modules used by multiple batch or TSO users concurrently, real storage working set requirements are greatly reduced because all users of a given module share the same copy.

Following are some considerations for placing Oracle modules in OS/390 link pack areas:

- An OS/390 IPL is generally required to add, remove, or replace a module in the OS/390 link pack areas. This complicates the timely application of maintenance or fixes.
- It might be necessary to move a module from the Oracle CMDLOAD library to another OS/390 data set so it is accessible for OS/390 link pack area placement.

- Adding modules to the PLPA reduces the maximum private area size of all OS/390 address spaces. This impact must be evaluated before moving new modules.

For details on adding modules to OS/390 link pack areas refer to the IBM documents for your platform and operating system. Details on which modules are candidates for OS/390 link pack area placement are covered in the *Oracle9i Enterprise Edition System Administration Guide for OS/390*.

Step 2: Examine Oracle Dump Data Sets and Modify as Necessary

When the gateway encounters an abend in one of its tasks, it dumps the abend to a SYS1.DUMP data set. Because the gateway does not attempt to dynamically allocate dump data sets, you must ensure a SYS1.DUMP data set is always available. The SYS1.DUMP data set must be large enough to hold two address spaces (the Oracle address space and the gateway address space). Refer to the IBM documents for your platform and operating system for information about managing dump data sets.

If a SYS1.DUMP data set is not available when needed, then OS/390 directs OSDI dumps to a SYSMDUMP DD statement if coded in OSDI Gateway startup JCL.

If a SYSMDUMP DD statement is directed to SYSOUT, then multiple dumps are preserved (at the cost of potentially large amounts of SPOOL space). The OS/390 external writer must be used to extract such dumps from the SPOOL file. This file can be written to a tape or DASD data set using an IBM external writer. For information about using the external writer program, refer to the IBM documents for your platform and operating system.

Do not specify SYSUDUMP and SYSABEND in OSDI Gateway startup JCL, because they produce dumps that are not computer-readable.

Step 3: Examine Oracle Trace Data Sets and Modify as Necessary

The gateway attempts to gather as much information as possible when internal errors occur for a user or process, and when gateway tracing is turned on via the TRACELEVEL parameter in the gateway environment PARMLIB member *sidENV*, where *sid* is the gateway service id. This information is placed in an Oracle trace data set. The trace data set name is generated based on the OSDI TRACE_DSNAME value defined in the OSDI Gateway Region parameters in PARMLIB member *sidPARM*, where *sid* is the gateway service id.

The TRACE_DSNAME value can specify either a SYSOUT specification or a data set name. For complete info and syntax of this parameter, see "[TRACE_DSNAME | TDSN](#)" on page 5-20.

Oracle Net for OS/390 supports network communications between Oracle applications and Oracle gateway systems across different OS/390 systems and different operating systems. Oracle provides OSDI listener (ORANET). This chapter describes how to configure it. For more information on Oracle Net, refer to the Oracle9i Net Services Book Set.

The following topics are included:

- [Overview](#) on page 6-2
- [OSDI Listener Architecture](#) on page 6-2
- [OSDI Listener Filenames](#) on page 6-3
- [Configuring the OSDI Listener](#) on page 6-4
- [Operating the OSDI Listener](#) on page 6-10
- [Formatting OSDI Listener Trace Files](#) on page 6-12
- [Oracle Advanced Security Option Encryption](#) on page 6-12

Overview

The OSDI listener (ORANET), also referred to as the Net service, runs as a service under an OSDI subsystem. In Oracle8i, Release 8.1.7 and Oracle9i, Release 1, all TCP and LU62 connections by Oracle applications, both client and server, were performed through the Net or Net8 service. Now, all Oracle clients on OS/390 open their own sockets.

The OSDI listener's primary function is to listen for inbound remote connections to an Oracle instance. For compatibility purposes, the OSDI listener still provides outbound connectivity services for Oracle9i, R1 and Oracle8i, 8.1.7 Oracle clients.

The OSDI Listener is not required for Oracle Net cross-memory protocol. The syntax of cross-memory protocol could be in either of these two forms:

Note that in the case of a database link from an OSI database instance to an OSDI gateway instance, the OSDI database instance is considered as an Oracle client in the context of OSDI listener for the Oracle gateway instance. Moreover, if the OSDI database instance resides in the same OS390 image as the OSDI gateway instance, Oracle Net cross-memory protocol could be used.

An OSDI Listener is not required for Oracle Net cross-memory protocol. The syntax of cross-memory protocol could be in either of two forms:

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=XM) (SUBSYS=ssn) (SERVICE=service_name) )
  (HS=) )
```

where *ssn* is the OSDI subsystem name and *service_name* is the OSDI gateway service name.

or:

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=XM) (SID=gateway_sid) )
  (HS=) )
```

where *gateway_sid* is the chosen OSDI gateway service SID.

Oracle recommends using the SID format for simplicity.

OSDI Listener Architecture

On OS/390, Oracle Net is implemented as an OS/390 OSDI service running in its own address space separate from the gateway service. The OSDI service acts as a

listener for the Oracle OS/390 instances and gateway instances. All protocol-specific code runs inside the OSDI listener.

Remote clients that access an OSDI service through an OSDI listener are dispatched on a lightweight unit of work called an enclave SRB. An enclave is created either once per session or for each SRB depending on the ENCLAVE keyword (described under "PARM" on page 6-5). An SRB is scheduled each time work is required to be done by the kernel. The enclave is deleted when the SRB completes. The OS/390 Workload Manager component may be used to control the execution characteristics of these enclave SRBs. Refer to "OS/390 Tuning" in Chapter 16, "Oracle9i Performance" of the *Oracle9i Enterprise Editing System Administration Guide for OS/390* for further details.

For client and server support, the OSDI listener uses the IBM macro implementation and a TCP/IP network to support network communications between the Oracle server and any remote OSDI listener TCP/IP client or server. For more information, refer to "[TCP/IP Network Considerations](#)" on page 6-8.

OSDI Listener Filenames

The product documentation, *Oracle Net Services Book Set*, refers to files in the following form:

basename.extension

where:

basename is the product name.

extension is the extension.

An example of this form is SQLNET.ORA.

These files are then converted to DDnames. The following DDnames are implemented under OS/390:

SQLNET defines a data set containing any SQLNET.ORA diagnostic, ASO, or Oracle names parameters. It is not necessary to allocate this DD unless these features are desired. Refer to *Oracle Net Services Book Set* or the *Advanced Security Administrator's Guide* for more information.

SQLNETTC	defines a data set into which trace output is written. It is recommended that this be defined as a SYSOUT data set in a held output class.
SQLNETLG	defines a data set into which any logging output is written. It is recommended that this be defined as a SYSOUT data set in a held output class.
TNSNAMES	defines a data set containing all the TNS connect descriptors and aliases for your installation. For further information on TNS connect descriptors, refer to the <i>Oracle Net Services Book Set</i> . This DDname is not necessary on server JCL unless DBLINKS originates from the server.
LDAP	defines the location of the LDAP server.
TNSNAV	TNS client navigation. (Generally not used on OS/390.)
INTCHG	Interchange. (Generally not used on OS/390.)

Example of diagnostic entries in SQLNET file

```

trace_file_client =/trace/sysout=x,hold
trace_file_server =/trace/sysout=x,hold
trace_file_agent =/trace/sysout=x,hold
trace_level_client = 16
trace_level_server = 16
trace_level_agent = 16
trace_functions_all = yes

```

Example of TNSNAMES entry for use by DBLINK definition

```

G4XXDB2 =
  (DESCRIPTION =
    (ADDRESS=(PROTOCOL=TCP) (HOST=your.host.tcpip.name)
      (PORT=port#) (SSN=net_sid))
    (HS=)
    (CONNECT_DATA=(SID=G4XX))
  )

```

Configuring the OSDI Listener

To create a Network Service under OSDI, you must first define the OSDI listener as a service using the OSDI DEFINE SERVICE command. In addition to defining the service, two other items that must be set up before the service can be started are: a JCL procedure, and network protocol-specific (TCP/IP) configuration. After you have defined OSDI listener as a service and have set up the additional items, you

ENCLAVE (SESS CALL)	specifies the duration of the enclave. When SESS is specified the enclave is created at logon and deleted at logoff. When CALL is specified the enclave is created when the server is sent a request, and is deleted when the server waits for a receive.
PORT (nnnn)	specifies the TCP/IP port number (<i>nnnn</i>) on which to listen for incoming connections. The default is 1521.
GTF	may be specified at the request of Oracle Support Services. This allows the OSDI listener internal trace to be captured to the OS/390 Generalized Trace Facility.
DUMP (nodename)	specifies the high level node, or nodes, of transaction dump data set names. The character string can be up to 26 characters in length, must follow the rules for OS/390 data set names, and must not end with a period. When an OSDI listener transaction dump occurs, then the value defined here will be prefixed to a string that includes a time and date stamp to generate a unique data set name. The default is ORACLE.TRANDMP.

Example of Network Service Definition

```
DEFINE SERVICE NET92 TYPE(NET) PROC(ORANET92) -
DESC(' Oracle Network Service 9.2') -
SID(NETG) -
PARM('HPNS GTF PORT91521) DUMP(ORACLE.TRANDMP)')
```

Note: The entire **PARM ()** string must be on one line.

Note: if you have an active OSDI database instance, you could use an existing OSDI subsystem for the Oracle gateway instance and/or use existing NET service instance for the Oracle gateway instance.

OSDI Listener Region JCL

As with a gateway service, a JCL procedure must be placed in a system procedure library prior to attempting a start of the service. The OSDI listener JCL procedure

name must have an associated OS/390 userid. Refer to the next topic, "[TCP/IP Network Considerations](#)" on page 6-8 for details. The EXEC card of the JCL must be equivalent to the following:

```
//NET      EXEC PGM=ORANET,REGION=0M
```

REGION=0M is specified to ensure that the service can allocate as much private virtual memory as it needs. Some OS/390 systems may prohibit or alter a REGION parameter such as this, so you might want to check with your systems programmer to determine if any changes must be made to allow the system to accept your REGION parameter. In addition, the following DD statements are required:

STEPLIB: This DD statement should point to the Oracle supplied 9.2 authload which contains the gateway and Net load modules.

NET8LOG: Connection-related informational messages, warning messages, and error messages are written to this sequential output file. Oracle Corporation recommends that it also be assigned to a JES spool file.

Note: The OSDI listener JCL procedure name must have an associated OS/390 userid. Refer to the next topic, "[TCP/IP Network Considerations](#)", for details.

Example of Network Service Procedure JCL

```
//NET EXEC PGM=ORANET,REGION=0M
//STEPLIB DD DSN= ORAN.ORAV.AUTHLOAD,DISP=SHR
//NET8LOG DD SYSOUT=X
```

Example of NET8LOG output

```
2000034 09:50:35.0 MIN0017I message service subtask initialized
2000034 09:50:35.0 MIN0016I command service subtask initialized
2000034 09:50:35.1 MIN0018I bind/unbind service subtask initialized
2000034 09:50:35.2 MIN0026I timer service subtask initialized
2000034 09:50:35.2 MIN0002I networking service NETC      initialization complete
2000034 09:50:35.2 MIN0005I global vector is at 19F0A000
2000034 09:50:35.2 MIN0024I connected to WLM subsystem OSDI
2000034 09:50:50.4 MIN0700I HPNS INITAPI call performed.  RC=0000, EC=00000
2000034 09:50:50.5 MIN0724I HPNS GHBY INITAPI call performed.  RC=0000, EC=00000
2000034 09:50:51.1 MIN0728I HPNS KID INITAPI call performed.  RC=0000, EC=00000
2000034 09:50:51.1 MIN0728I HPNS KID INITAPI call performed.  RC=0000, EC=00000
2000034 09:50:51.1 MIN0728I HPNS KID INITAPI call performed.  RC=0000, EC=00000
```

```
2000034 09:50:51.2 MIN0728I HPNS KID INITAPI call performed. RC=0000, EC=00000
2000034 09:50:51.2 MIN0728I HPNS KID INITAPI call performed. RC=0000, EC=00000
2000034 09:50:51.2 MIN0728I HPNS KID INITAPI call performed. RC=0000, EC=00000
2000034 09:50:51.2 MIN0713I I am listening on port 01522 socket 00000
2000034 10:05:58.8 MIN0733I Socket 0000 connected Subtask Kid1, IP 144.025.040.217, Port 01129.
2000034 10:05:58.8 MIN0733I Socket 0000 connected Subtask Kid2, IP 144.025.040.217, Port 01130.
2000034 12:00:13.9 MIN0098I networking service NETC      termination in progress
2000034 12:00:18.9 MIN0722I HPNS Kid #003 shut down.
2000034 12:00:18.9 MIN0722I HPNS Kid #001 shut down.
2000034 12:00:18.9 MIN0722I HPNS Kid #006 shut down.
2000034 12:00:18.9 MIN0722I HPNS Kid #002 shut down.
2000034 12:00:18.9 MIN0722I HPNS Kid #005 shut down.
2000034 12:00:18.9 MIN0722I HPNS Kid #004 shut down.
2000034 12:00:18.9 MIN0723I HPNS Gethostbyname subtask ended.
2000034 12:00:18.9 MIN0721I HPNS shut down, GoodBye.
2000034 12:00:18.9 MIN0091I timer service subtask terminated
2000034 12:00:18.9 MIN0095I bind/unbind service subtask terminated
2000034 12:00:18.9 MIN0093I command service subtask terminated
2000034 12:00:18.9 MIN0094I message service subtask terminated
MIN0000I End of Net8 Log.
```

TCP/IP Network Considerations

The OSDI Listener uses the MACRO API interface for TCP/IP, and distributes the communications processing workload across multiple tasks in the OSDI listener address space.

If the IBM stack is being used, then particular attention must be paid to the MAXFILEPROC and MAXSOCKETS parameters (under AF_INET) in the BPXPRMxx member of SYS1.PARMLIB. These parameters must be set high enough to support the expected connection load. Both of these parameters can limit the number of connections that the OSDI listener will be able to open. Also, the OSDI listener JCL procedure name must have an associated OS/390 userid in order to use TCP/IP, which is controlled by OS/390 UNIX System Services. The userid must have an OMVS RACF segment (or equivalent, if a product other than RACF is used) if the installation is not using a default OMVS segment.

In addition, the interface resolves names through the standard GETHOSTBYNAME API. Thus the resolution depends on how IBM TCP/IP is configured. If a DNS is defined to TCP/IP, then it will be used. Otherwise, TCP/IP will default the processing to its SITEINFO file. Also, IBM's Language Environment runtime library (LE) must be available through a STEPLIB DD or linklist to the OSDI listener address space in order for GETHOSTBYNAME to work. This is an IBM requirement. TNS does a GETHOSTBYNAME call at startup to test the function. This call may take

minutes to complete if a busy name server is involved. The interface is not ready for work until the MIN0713I message is displayed on the system console. For more information on the GETHOSTBYNAME API, refer to the relevant IBM documentation on TCP/IP.

Client-Server Access Using the OSDI Listener

Note: In this section, the term 'client' refers to the Oracle integrating server in the context of a database link session to an OSDI gateway instance.

Remote Clients

Remote (inbound) clients access Oracle gateway instances through the OSDI listener as follows:

1. The OSDI network service listens on a single endpoint (network address) for each protocol. All remote clients that go through a particular OSDI listener with a particular protocol use the same network address regardless of which gateway instance they want to access. All TCP/IP clients specify the same hostname (or IP) and port number.
2. Clients indicate the target gateway instance that they want with the ' (CONNECT_DATA= (SID=SSSS)) ' clause in the OSDI listener address string.

The following is an example of a tnsname entry for a database link from a remote Oracle database server to an OSADI gateway instance through TCP/IP.

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=TCP) (HOST=MVS08) (PORT=1999) )
  (HS=)
  (CONNECT_DATA= (SID=TGD4) )
)
```

MVS08 is the hostname that the gateway resides in. 1999 is the port number that OSDI NET listens on that serves the gateway instance. TGD4 is the OSDI TG4DB2 SID it's trying to connect to.

Note that if the remote Oracle database server is an OSDI database instance of release 9.1 or below, you would need to specify SSN parameter (within the ADDRESS specification) that specifies the OSDI NET SID that serves the outbound traffic for the OSDI database instance.

For MPM and OSDI compatibility, refer to Chapter 11.

Oracle clients on OS/390 are also able to use an Oracle Names or LDAP server running on another platform to resolve connection requests. The following samples of the OSDI listener configuration file are required to make use of this service.

Name Server

SQLNET DD or SQLNET.ORA Definitions

```
#####
# Names .....: (CONNECT_TIMEOUT = 0) -MUST- be specified
#####
NAMES.DEFAULT_DOMAIN = world
NAMES.DEFAULT_ZONE = my.domain.com
NAMES.DIRECTORY_PATH = (TNSNAMES,ONAMES,LDAP)
NAMES.PREFERRED_SERVERS =
  (ADDRESS_LIST =
    (DESCRIPTION =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = names_host)
        (Port = 1575)
      )
      (CONNECT_TIMEOUT = 0)
    )
  )
##
```

LDAP Server

LDAP DD or LDAP.ORA Definitions

A sample LDAP.ORA file:

```
DEFAULT_ADMIN_CONTEXT = "c=us"
DIRECTORY_SERVERS = (hostname:389:636)
DIRECTORY_SERVER_TYPE = OID
LDAP.ORA can be generated using the NETCA utility.
```

Operating the OSDI Listener

The OSDI listener is started by the OSDI subsystem start command, for example:

```
ORSS START NET
```

This command would start the OSDI listener defined in the earlier example for "Example of Network Service Definition" on page 6-6 if the subsystem were named 'ORSS'. You should then see the OSDI listener PROC start up followed by the following messages from the OSDI listener address space:

```
MIN0001I networking service initializing
MIN0002I networking service NET8      initialization complete
MIN0713I I am listening on port 01521 socket 00000
```

Additional messages are written to the NET8LOG DD, but message traffic to the console is limited to error and warning messages.

Several commands are available for communicating with a running Net service. Commands are issued using the OS/390 MODIFY (or F) system operator command with the general format:

```
F name,cccc pppppp
where:
```

- name* is the jobname or identifier of the OSDI listener
- cccc* is a command verb from the table below
- pppppp* represents an appropriate parameter for that command

Table 6–1 Command Verbs for OS/390 MODIFY (or F) System Operator Command

start	hpns	Starts support for the specified protocol in the OSDI listener.
stop	hpns	Stops support for the specified protocol.
dis	tcp all pool	Display information about existing connections for the specified protocol or storage pool statistics.

The OSDI listener can be stopped with the OS/390 stop command (STOP or P), as in 'p net', or via the OSDI subsystem stop command, as in 'ORSS STOP NET'. In either case, the following messages will be seen on the console:

```
MIN0098I networking service NET termination in progress
MIN0721I HPNS shut down, GoodBye.
MIN0099I networking service termination complete
```

The OSDI listener service will also respond to the OSDI subsystem 'display' and 'display long' commands with appropriate information from the address space. Finally, the OSDI subsystem 'drain' command will prevent any new connections on either protocol. Existing connections will not be affected. The OSDI subsystem 'RESUME' command will restore the ability of clients to establish new connections through the OSDI listener.

Formatting OSDI Listener Trace Files

The OSDI listener provides a utility program called TRCASST that formats the trace files OSDI listener can produce. You may be asked to run TRCASST to help gather diagnostic information required by Oracle Support Services. Sample JCL for TRCASST is provided in oran.orav.SRCLIB(TRCASST).

Before you use TRCASST, ensure that the trace files have not been created with carriage control. TRCASST will be unable to process such files.

When TRCASST runs, the TNSUSMSG DDname must point to a PDS containing a TNSUS message file. This file was placed into oran.orav.MESG(TNSUS) during OSDI listener installation.

Oracle Advanced Security Option Encryption

The OSDI listener supports CHECKSUM and encryption algorithms. The following sections describe a basic method of verifying this feature, if it is to be used by your site. The easiest way to tell if Oracle Advanced Security Option (ASO) 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. ASO encryption is working properly if no further error messages are received.

The following procedures test ASO encryption by this method. The incorrect parameter settings produce error 12660.

Setting Up ASO Encryption for Test

Perform the following steps to set up ASO encryption:

Checklist for Setting Up ASO Encryption

1. Set ASO encryption parameters for the server.
2. Set ASO encryption parameters for the client.

Step 1: Set ASO Encryption Parameters for the Server

Use ISPF to edit the OSDI listener configuration file on the OS/390 system (server system) to add the following parameters and values. If the server is remote (not OS/390), then use the appropriate editor for the server platform to change SQLNET.ORA.

```
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 = "abcdefg"
```

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 ASO Encryption Parameters for the Client

Edit the OSDI listener configuration file on the client 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 = "abcdefg"
```

The value shown for `SQLNET.CRYPTO_SEED` is only an example. Set it to the same value used on the server system.

Testing ASO Encryption

After completing Steps 1 and 2 of the configuration procedure, you are ready to test the operation of the ASO encryption.

Checklist for Testing ASO Encryption

1. Connect client and server.

2. Reset configuration parameters on server.

Step 1: Connect Client and Server

Attempt a connection between the server and client systems. You should receive the following error message:

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

Step 2: Reset Configuration Parameters on Server

Change the ASO encryption parameters on the server to:

```
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUIRED  
SQLNET.ENCRYPTION_SERVER = REQUIRED
```

Attempt the connection between the client and server again. If no error message is returned and the connection completes, then ASO encryption is working properly.

Administering the Gateway

This chapter describes the basic administration tasks for Oracle Transparent Gateway for DB2, including implementing security strategies and enabling diagnosis and error reporting. Refer to [Appendix B, "OSDI Subsystem Command Reference"](#), for the subsystem administration tasks requiring OSDI commands and parameters.

The following topics are included:

- [Operation of the Gateway Subsystem with OSDI](#) on page 7-2
- [Controlling Access to OSDI Subsystem Commands](#) on page 7-2
- [Controlling Access to OSDI Services](#) on page 7-4
- [Gateway Security](#) on page 7-4
- [SAF Router Considerations](#) on page 7-5
- [Gateway User Exit Facility](#) on page 7-5
- [DB2AUTH Exit](#) on page 7-8
- [Sample Exit Programs](#) on page 7-9

Operation of the Gateway Subsystem with OSDI

Starting and stopping the gateway service is accomplished using OSDI operating commands. They are normally issued using an OS/390 system operator command interface. The target subsystem is specified by using the command prefix associated with the subsystem, or the subsystem name. All of the operating commands take a gateway service name as the first positional parameter. This service name must be the name of a defined service.

START

The START command initiates execution of an address space for a specified service. For a gateway service, the service must not already be running.

The command structure for starting a gateway service is shown in the following example:

```
/ssn START name [ PARM( string ) ]
```

Refer to [Appendix B, "OSDI Subsystem Command Reference"](#) for complete syntax of the START command.

STOP

The STOP command requests termination of a running service. The normal mode of stop changes the service state to stopping and posts the stop request to the service; it is up to the service to comply, presumably after allowing current requests to complete and performing required cleanup. A force option causes the service address space(s) to be terminated involuntarily. The force form of stop requires that a normal stop be issued first. This command has no effect when the service is not running.

The command structure for stopping a service is shown in the following example:

```
/ssn STOP name [ FORCE ]
```

Refer to [Appendix B, "OSDI Subsystem Command Reference"](#) for complete syntax of the STOP command.

Controlling Access to OSDI Subsystem Commands

OSDI subsystem command processing includes an authorization check to confirm that the console or the user is allowed to issue the command. You control access to commands by defining resource profiles to the security subsystem and then by granting access (for specific consoles and users) to those resources. If you do not

define resource profiles, then the authorization check returns a "resource unknown" indication to OSDI, and OSDI then allows the command to be processed. Thus, the default behavior (in the absence of any profile definitions) is that any command is allowed from any source. This is not chaotic, as it may sound, because access to command-issuing mechanisms themselves (such as consoles) is usually controlled in most OS/390 installations. Base your decision to define profiles and to activate the authorization mechanism upon the security standards and procedures at your installation.

The resource profiles that are used to protect commands should be defined in the resource class that you specified in the cmd-class field of the INIT record in the subsystem parameter file. If you elected to accept the default, then this will be the FACILITY class. Otherwise, it will be a class name that you chose and configured for your SAF-compliant security software.

The command authorization resource names are of the form:

ssn.cmdverb

where ssn is the OSDI subsystem name, and cmdverb is the full-length OSDI command verb. (Verb abbreviations, such as DEF and ALT, must not be used in the resource name.)

The level of authorization that must be granted to users or to consoles in order to enable commands depends on the command. The following table lists all of the command verbs and the authorization level required for each:

Table 7-1 Command Verbs and their Authorization Level

Verb	Authorization Level
DEFINE	Update
ALTER	Control
SHOW	Read
START	Read
DISPLAY	Read
DRAIN	Read
RESUME	Read
STOP	Read

Controlling Access to OSDI Services

OSDI bind processing, which establishes connections between OS/390 address spaces, performs an authorization check to confirm that the binding address space (the "client") is allowed to access the target service.

In the case of Oracle gateways, the only potential sensible case is that "the client" is the OSDI integrating database server and the target service is the Oracle gateway.

In order for the check for a given target service to be meaningful, resource profiles must be defined to a SAF-compliant security server such as RACF. The profile names incorporate the OSDI service name so that access to each service is separately controlled. When profiles are defined, the OS/390 userid that is associated with the client address space must have READ authorization on the target service's profile in order for the bind to be allowed.

If you do not define resource profiles for a service, then all binds from all address spaces are permitted. Oracle Corporation recommends that you define resource profiles for all services so that bind access is controlled via standard OS/390 security mechanisms.

The resource profiles that are used to protect binds should be defined in the resource class that you specified in the bind-class field of the INIT record in the subsystem parameter file. If you elected to accept the default, then this will be the FACILITY class. Otherwise it will be a class name that you chose and configured for your SAF-compliant security software.

The name structure for the managed binds used by Oracle database links is"

ssn.service.ABIND

where ssn is the OSDI subsystem name, service is the target service name (which is the gateway), and a BIND is a constant indicating managed binds.

Gateway Security

The Oracle userid and password is passed over the database link to the gateway to authorize gateway users to DB2 objects. If the CONNECT TO clause is specified when creating the database link, then the userid and password sent to the gateway are those specified in this clause. If the CONNECT TO clause is omitted from the database link specification, then the Oracle userid and password using the database link are passed to the gateway for authorization.

When the Oracle gateway receives the userid and password from the database link, a security environment is established that is local to the OS/390 and DB2 systems. The following steps occur when establishing a gateway security environment:

1. A gateway security user exit is called by the gateway to verify the database link userid and password is valid for OS/390, and to set up the security environment for the gateway. The sample user exit delivered with the gateway is called G4AUTH. One of the tasks of G4AUTH is to create a structure that contains the DB2 primary and secondary authorization ids. This is used by DB2 when it calls the DSN3@ATH user exit.
2. The gateway initiates a connection to DB2 using a DB2 CAF OPEN. Subsequently, DB2 calls its connection exit, DSN3@ATH. All DB2 users, including non-gateway users, connecting to DB2 through CAF OPEN and CONNECT cause DB2 to call this security exit. The gateway provides a sample exit to replace the DB2 user exit, DSN3@ATH, so the primary and secondary userids can be retrieved by DB2 from the structure created by G4AUTH. The linkedit of DB2AUTH links in the site's existing DSN3@ATH so that if the Oracle specific code determines that the connecting user is not from the gateway, it will fall through to the pre-existing logic.

The remaining sections of this chapter provide more specific information about gateway security, including:

- OS/390 Security Authorization Facility (SAF) considerations
- Gateway User Exit Facility
- Gateway DSN3@ATH replacement (DB2AUTH)

SAF Router Considerations

The gateway uses the SAF router to validate userids and passwords. It is also used to check user access to the DB2 security profile, with CLASS set to DSNR and ENTITY set to db2_subsys.BATCH. The DB2 security profile is optional.

Security environments using the SAF interface include ACF2, RACF, and TOP SECRET. If your local security environment does not use the SAF interface, then your system programmer can replace the SAF calls where appropriate.

Gateway User Exit Facility

The gateway user exit facility provides gateway installations with a mechanism for passing control to installation-written code at user logon time.

If any user exit module is not part of the gateway or OSDI, then it is an OS/390 load module (exit module). During OSDI startup, these user exit modules are loaded by OSDI dynamically into the gateway address space from one of these locations:

- The OSDI STEPLIB
- The system linklist (LNKLST00)

Specifying an Exit Module

Exit modules are specified to the gateway with the gateway region parameter, LOGON_AUTH, which is accessed at OSDI initialization time:

LOGON_AUTH (name)
where:

name

is the exit module member name.

G4AUTH is supplied to provide the standard SAF security check. Optionally, ALLAUTH can be used to skip the SAF security check.

To display the status of an EXIT, use the OSDI DISPLAY EXIT command.

The logon exit must reside in the STEPLIB or JOBLIB concatenation or in the linklist, and it must be in an authorized library.

A sample user logon exit is provided in gateway SRCLIB library member G4AUTH. It uses the OS/390 SAF interface to invoke the OS/390 security manager. If the OS/390 security manager does not support the SAF interface, then the calls to RACROUTE in the exit must be replaced with the equivalent calls appropriate for the OS/390 security manager.

The calling sequence for the logon exit uses standard OS/390 assembler calling conventions. R15 is the entry point, R14 is the return address, R13 points to a standard 72-byte save area, and R1 is the address of a parameter list. The parameter list consists of a list of addresses of each parameter (all values are passed by reference, not by value), and the last parameter has the high-order bit set.

When returning, R15 should be set to 0 to indicate a successful verification of the userid and password that were supplied, and should be set to any nonzero value to indicate any type of failure (4 would be an appropriate value).

The exit is called in 31-bit addressing mode, supervisor state, storage protection key 7, and in an authorized address space. The exit will be running in TCB mode with

no locks held and with no ARRs, FRRs, or EUT-style FRRs set. The exit is called in primary addressing mode with HASN=PASN=SASN (home, not cross memory mode).

The logon exit should be fully reentrant code.

The parameter list that is passed contains pointers to the following parameters, all of which are input only:

Table 7-2 Gateway Region Parameters

Field	Type/Length	Description
work area	char/4k	set to all x'00' before every call to the exit
userid	char/1+	userid to be validated (number of bytes varies)
userid length	bin/2	length of userid
password	char/1+	password to be validated (number of bytes varies)
password length	bin/2	length of password
OS/390 jobname	char/8	OS/390 jobname from JOB card of client
ASID	bin/2	address space id of client address space
OSDI session id	bin/4	a unique OSDI session id
OS username	char/8	the operating system username (batch, TSO, CICS, and IMS only)
terminal name	char/8	terminal name
program name	char/8	program name
RACF group name	char/8	RACF group name
connection type	char/8	connection type (BATCH, TSO, CICS, IMS, TP/IP, VTAM) to indicate environment of client
JES jobid	char/8	JES job identifier (such as JOB08237)
job card entry time	bin/4	entry (submission) time of job. Binary hundredths of a second since midnight
job card entry date	packed/4	entry (submission) date of job. Packed decimal 0CYDDDF, where C=0 for 19, C=1 for 20, YY=year, DDD=day number within the year (Jan 1=1)

Table 7–2 Gateway Region Parameters

Field	Type/Length	Description
job card accounting info	char/145	from jobcard
network data (high bit set in parameter list)	char/2+	variable length NIV data (refer to Chapter 9, "Oracle SMF Data")

The only output of the logon exit is the R15 return code. No other value that is passed in the parameter list should be modified except the first one.

The first parameter is a 4096-byte work area that is set to all x'00' before every call to the logon exit. The logon exit can use this storage for anything that it needs. It should not be freed.

The logon exit can do any of the following:

- call a security manager (SAF calls, RACF, Top Secret, ACF2, and so forth)
- get and free storage using the STORAGE macro (the exit must keep track of all acquired storage and must make certain to free it before exiting)
- call SMF to write SMF records
- call WTO to write messages to the console

The logon exit should not do anything that would cause it to wait for any significant period of time (more than one tenth of a second, for example). Avoiding opening data sets, writing to the operator with a reply (WTOR), and creating enqueues.

Any resources that are acquired in the logon exit must be freed before it returns. There is no cleanup call made to the logon exit, so any resources that are not released will accumulate in the address space and could eventually cause resource shortages.

DB2AUTH Exit

The DB2 connection exit (DSN3@ATH) is called by DB2 in response to CAF CONNECT and OPEN calls from gateway and non-gateway users. For additional information on the DB2 connection exit, refer to the IBM documents for your platform and operating system.

The gateway implementation of DSN3@ATH is in the SRCLIB library in member DB2AUTH. The linkedit for DB2AUTH includes the previously existing copy of DSN3@ATH.

If DB2AUTH determines the user is not connected from a gateway subsystem, then control is passed to the previously existing copy of DSN3@ATH. This copy of DSN3@ATH does its own security checking.

The previously existing DSN3@ATH is:

- The standard version shipped with DB2, or
- Some other, tailored version of DSN3@ATH

The gateway verifies it is checking security on a gateway user before it proceeds. It does not conflict with standard DB2 security checking.

The DB2AUTH security mechanism works as follows:

1. When DB2 is started, DB2 loads DSN3@ATH into memory.
2. At logon time, the gateway calls the G4AUTH exit to verify the user's userid and password.
3. The G4AUTH exit creates or reuses the LOGDATA structure, and initializes it with the userid and a list of secondary ids.
4. The G4AUTH exit returns control to the gateway.
5. The gateway displays a CAF OPEN call to DB2.
6. DB2 passes control to its connection exit, DSN3@ATH, which is the DB2AUTH exit. DSN3@ATH runs on the same TCB and address space as the caller of the CAF OPEN (in this case, the gateway).
7. DB2AUTH copies userids from the LOGDATA structure to the DB2 AIDL structure.
8. DB2AUTH passes control back to DB2.
9. DB2 passes control back to the gateway.

Sample Exit Programs

Sample Exit Programs are supplied in the SRCLIB dataset.

Using the Gateway

Using the gateway involves connecting to the gateway system and the remote DB2 database associated with it.

The following topics are included:

- [Database Link Behavior](#) on page 8-1
- [Managing Threads](#) on page 8-4
- [Gateway CPU Time](#) on page 8-5
- [Using DB2 Cursors](#) on page 8-5
- [Using the Synonym Feature](#) on page 8-6
- [Read-Only Gateway](#) on page 8-6
- [Performing Distributed Queries](#) on page 8-7
- [Replicating in a Heterogeneous Environment](#) on page 8-8
- [Copying Data from the Oracle Database Server to the DB2 Server](#) on page 8-9
- [Copying Data to the Oracle Database Server from the DB2 Server](#) on page 8-14

Database Link Behavior

A connection to the gateway is established through a database link when it is first used in a gateway session or transaction. In this context, connection refers to the connection between the Oracle database server and the gateway. The connection remains established until the session ends. Another session or user can access the same database link and get a connection to the gateway and DB2 database.

Connections to the DB2 database might be limited by factors that include memory, gateway parameters, or DB2 server resources.

The database and application administrators of a distributed database system are responsible for managing the necessary database links defining paths to the gateway.

Database links are discussed in detail in the *Oracle9i Administrator's Guide*. Information for using database links with the gateway is discussed here.

Creating Database Links

To create a database link and to define a path to the gateway, use the CREATE DATABASE LINK statement. The CONNECT TO clause specifies the remote userid and password to use when creating a session in the gateway. If you do not specify a userid and password in the CONNECT TO clause, then the Oracle logon userid and password are used. The USING clause specifies a TNSNAMES.ORA connect descriptor.

Creating Database Links Using Oracle Net

Oracle Net is required. The following syntax creates a database link to access information in the DB2 database using Oracle Net:

```
CREATE DATABASE LINK dblink
    CONNECT TO userid identified by password
    USING 'tns_name_entry';
```

where:

<code>dblink</code>	is the complete database link name (such as gateway).
<code>userid</code>	is the userid used to establish a session in the remote database. It must be authorized to any table or file on the DB2 server referenced in the SQL commands. The userid cannot be longer than eight characters.
<code>password</code>	is the password used to establish a session in the remote database. This must be a valid DB2 server password. The password cannot be longer than eight characters.
<code>tns_name_entry</code>	specifies the Oracle Net TNS connect descriptor used to identify the gateway instance. For an example of a tnsnames entry, refer to Chapter 6.

Guidelines for Database Links

Once used, a database link remains open for the duration of the gateway session. If you want to close a database link during a session, then you can do so with the `ALTER SESSION CLOSE DATABASE LINK` statement.

Accessing Data through Database Links

DB2 tables, views, synonyms, and aliases available to the userid specified in the `CONNECT TO` clause can be accessed with the following syntax:

```
SELECT * FROM SCOTT.EMP@gateway
```

The `CONNECT TO` userid provides implicit qualification for unqualified tables. For example:

```
SELECT * FROM EMP@gateway
```

resolves to `SCOTT.EMP` on DB2 if the `CONNECT TO` user is `SCOTT`. If no `CONNECT TO` statement is defined with the database link, then the Oracle userid using the database link is used as the implicit qualifier.

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 following statement:

```
DROP DATABASE LINK dblink;
```

Do not drop a database link 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. The `USER_DB_LINKS` data dictionary view shows the database links defined for a specific Oracle user. The `ALL_DB_LINKS` data dictionary views show all defined database links both public and private. The user has access to all these views. The `DBA_DB_LINKS` dictionary view, accessible only to users with DBA authorization, shows all database links defined in the gateway instance.

Limiting the Number of Active Database Links

You can limit the number of connections from a user process to remote databases with the `INIT.ORA` parameter `OPEN_LINKS`. This parameter controls the number of remote connections any single user process can use concurrently with a single SQL statement. Refer to the *Oracle9i Administrator's Guide* for additional information about limiting the number of active database links.

Managing Threads

Whenever a client connects to the Oracle database server to access data from a DB2 server, the Oracle database server creates and manages database access threads between the client and the Oracle database server. DB2 creates and manages allied threads between the gateway and DB2. The Oracle system manages the threads between the client and the Oracle database server. The DB2 system manages the threads between the gateway and DB2.

When using the gateway to access DB2 data in a client/server configuration, you can encounter the following scenarios:

1. A user turns off a workstation abnormally and the thread connection remains active.
2. A user leaves a workstation in an idle state for an extended period of time and the thread connection remains active.
3. A user or application uses a workstation to enter a long-running query that maintains a lock in DB2 and, thus, an active thread.

When connecting to DB2 through the gateway, it is important to remember that, if the client or server is abnormally terminated, a connection can be left open indefinitely, unless specifically identified and closed by the system.

KEEPALIVE

The `KEEPALIVE` functionality can be used to resolve scenario one mentioned previously. For Oracle systems using `KEEPALIVE`, the identification of an inactive client/server connection is handled differently in the UNIX environment from the way it is handled in the OS/390 environment.

With UNIX, an optional parameter `SQLNET.EXPIRE_TIME` in the `SQLNET.ORA` file determines how often Oracle Net sends a probe to verify whether a client/server connection is still active. If the connection is inactive, then the Oracle

database server cleans up the connections between the client, the Oracle database server, the gateway, and DB2.

With OS/390, KEEPALIVE sends a probe to verify whether a client/server connection is still active. The KEEPALIVE functionality is implicitly leveraged by the individual protocol vendors. For example, if you are using the TCP/IP protocol and KEEPALIVE is enabled, then the KEEPALIVE functionality is used automatically by Oracle Net for OS/390.

Canceling DB2 Threads

The DB2 command CANCEL THREAD can potentially be used to alleviate problems in scenario three by scheduling threads to be terminated. The user or application must still attempt to access DB2 again before the thread can be terminated.

For further information on this feature, refer to the IBM documents for your platform and operating system.

Gateway CPU Time

The Oracle Transparent Gateway for DB2 uses the CAF to connect to the target DB2 system. When the gateway is used to access data in DB2, much of the DB2 SQL processing takes place in cross memory mode, running under a TCB in the gateway address space with the CAF. In turn, the CPU time charged to the gateway address space includes both CPU utilization of the gateway and the CPU time required for DB2 SQL processing. Thus, the SMFXMCPUR field, which records CPU utilization, is a combination of the gateway and the DB2 CPU time. Therefore, this field does not represent pure gateway CPU time and is not a good predictor for judging gateway efficiency.

Using DB2 Cursors

The maximum number of DB2 cursors the gateway can open per Oracle session is 5000. Although the gateway can open 5000 cursors, other Oracle database server or DB2 limits might affect how many cursors can actually be opened for a specific application. The default is 50.

Ensure the HS_OPEN_CURSORS parameter in member *sidENV* of the PARMLIB library is set to the maximum you require.

Using the Synonym Feature

You can provide complete data, location, and network transparency by using the synonym feature of the Oracle database server. When a synonym is defined, you do not need to know the underlying table or network protocol being used. A synonym can be public, which means all users can make reference to the synonym. A synonym can also be defined as private, which means every user must have a synonym defined to access a DB2 table. Refer to the Oracle database server documentation for details on the synonym feature.

The following statement creates a system wide synonym EMPDB2 for the EMP file in the DB2 server Oracle library:

```
CREATE PUBLIC SYNONYM EMPDB2 FOR SCOTT.EMP@gateway
```

Only those with database administrator authority can create public synonyms. You can use a similar statement to create a private synonym if you do not have database administrator authority:

```
CREATE SYNONYM EMPDB2 FOR SCOTT.EMP@gateway
```

Read-Only Gateway

The read-only option can provide improved performance and security based on your configuration and parameter selections. An environment parameter, DB2READONLY, 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 DB2. 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 in DB2 are rejected.

Oracle Corporation recommends that you do not routinely switch between settings of the DB2READONLY parameter. If you need both update and DB2READONLY functionality, then you should install two separate instances of the gateway with different read-only settings.

If your system can tolerate an occasional dirty read, then you can bind the gateway plan using the isolation level (uncommitted read). This eliminates DB2 locking problems and improves overall performance.

Performing Distributed Queries

The gateway technology enables the completion of distributed queries joining data from the Oracle database server and the DB2 server, and any other data store for which Oracle Corporation provides a gateway. These complex operations are transparent to the users requesting such data retrieval.

Example of a Distributed Query

The following example joins data between the Oracle database server and multiple DB2 servers:

```
SELECT O.CUSTNAME, P.PROJNO, E.ENAME, SUM(E.RATE*P.HOURS)
FROM ORDERS@GATEWAY_1 O, EMP@ORACLE9 E, PROJECTS@GATEWAY_2 P
WHERE O.PROJNO = P.PROJNO
AND P.EMPNO = E.EMPNO
GROUP BY O.CUSTNAME, P.PROJNO, E.ENAME;
```

Through a combination of views and synonyms, using the following SQL statements, the process of distributed queries is made transparent to the user:

```
CREATE SYNONYM ORDERS FOR ORDERS@GATEWAY_1;
CREATE SYNONYM PROJECTS FOR PROJECTS@GATEWAY_2;
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;
```

With the views and synonyms in place, the user retrieves information from these three data stores in one command:

```
SELECT * FROM DETAILS;
```

which produces the following:

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

CUSTNAME	PROJNO	ENAME	SPEND
XYZ Inc.	2	Smith	180

Two-Phase Commit Processing

The gateway must coordinate the distributed transaction and only one gateway can participate in an Oracle two-phase commit transaction.

Two-phase commit transactions are recorded in the DB2 table ORACLE2PC, which is created during gateway installation.

On all systems, the ORACLE2PC table must be available at all times. For security reasons, users must not have direct access to this table. The table is accessed and updated by the gateway internally.

Distributed DB2 Transactions

Because the ORACLE2PC table is used to record the status of a gateway transaction, the table must reside in the database where the DB2 update takes place. Updates to the ORACLE2PC table cannot be part of an IBM distributed transaction.

Replicating in a Heterogeneous Environment

Oracle Transparent Gateway for DB2 provides a number of options for replicating Oracle and non-Oracle data throughout the enterprise.

Oracle Database Server Triggers

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

Oracle Snapshots

Oracle Transparent Gateway for DB2 can use the Oracle snapshot feature to automatically replicate non-Oracle data into the Oracle database server. This complete refresh capability of Oracle snapshots can be used to propagate a complete copy or a subset of the non-Oracle data into the Oracle database server at user-defined intervals.

Copying Data from the Oracle Database Server to the DB2 Server

Data can be copied from the Oracle database server to the DB2 server by two methods:

- Triggers
- SQL*Plus COPY command

Triggers

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

For example, you have an Oracle ORA_EMP table that contains ENAME and EMPNO. You also have a table called DB2_EMP, which is a copy of ORA_EMP and which resides on DB2. You want all changes made to the Oracle ENAME to be reflected immediately in your DB2_EMP table on DB2. In this scenario, an Oracle database server trigger can be developed to run every time an update is made to ENAME in your Oracle ORA_EMP table:

```
CREATE OR REPLACE TRIGGER EMP_TRIGGER
  AFTER UPDATE OF ENAME ON SCOTT.ORA_EMP
  FOR EACH ROW
BEGIN
  UPDATE SCOTT.DB2_EMP@tg4db2
  SET ENAME = :NEW.ENAME
  WHERE EMPNO = :NEW.EMPNO;
END;
```

where *tg4db2* is the name of the database link used to access the gateway.

SQL*Plus COPY Command

The SQL*Plus COPY command copies data from the Oracle database server to the DB2 server. The SQL command INSERT is not supported. The following command:

```
INSERT INTO gateway_table SELECT * FROM oracle_table;
```

displays the following message:

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

Use the following SQL*Plus syntax to copy data from your local Oracle database server to the DB2 server:

```
COPY FROM username/password@ORACLE9
INSERT destination_table
USING query;
```

The next example selects all rows from the local Oracle EMP table and inserts them into the EMP table on the DB2 server:

```
COPY FROM SCOTT/TIGER@ORACLE9
INSERT SCOTT.EMP@gateway
USING SELECT * FROM EMP;
```

Notes: The SQL*Plus COPY command supports APPEND, CREATE, INSERT, and REPLACE options.

However, INSERT is the only option supported when copying to the DB2 server.

For more information about the COPY command, refer to the *SQL*Plus User's Guide and Reference*.

STREAMS Replication

TG4DB2 and Heterogeneous Services now support replication to DB2 using Oracle streams. Oracle Streams is a rule base process which allows changes to an Oracle table to be captured and applied to an equivalent DB2 table based upon user written rules. Replication from DB2 to Oracle is not supported at this time.

Before setting up your Streams Replication environment ensure that archivelog is enabled, nothing else will work.

An example of a simple table replication follows:

First you should grant the necessary authorizations to your Streams admin userid.

```
CONNECT SYS/SYS_PASSWORD AS SYSDBA

GRANT CONNECT, RESOURCE, SELECT_CATALOG_ROLE
  TO strmadmin IDENTIFIED BY strmadminpw;

GRANT EXECUTE ON DBMS_APPLY_ADM TO strmadmin;
GRANT EXECUTE ON DBMS_AQADM TO strmadmin;
GRANT EXECUTE ON DBMS_CAPTURE_ADM TO strmadmin;
```

```

GRANT EXECUTE ON DBMS_FLASHBACK TO strmadmin;
GRANT EXECUTE ON DBMS_PROPAGATION_ADM TO strmadmin;
GRANT EXECUTE ON DBMS_STREAMS_ADM TO strmadmin;

BEGIN
  DBMS_RULE_ADM.GRANT_SYSTEM_PRIVILEGE(
    privilege => DBMS_RULE_ADM.CREATE_RULE_SET_OBJ,
    grantee => 'strmadmin',
    grant_option => FALSE);
END;
/

BEGIN
  DBMS_RULE_ADM.GRANT_SYSTEM_PRIVILEGE(
    privilege => DBMS_RULE_ADM.CREATE_RULE_OBJ,
    grantee => 'strmadmin',
    grant_option => FALSE);
END;
/

```

Then set up the Streams queue and the database link the apply process will use.

```

CONNECT strmadmin/strmadminpw

EXEC DBMS_STREAMS_ADM.SET_UP_QUEUE();

DROP DATABASE LINK strmdblink.your.domain.com;

CREATE DATABASE LINK strmdblink.your.domain.com
  CONNECT TO userid IDENTIFIED BY password
  USING 'tnsnames_entry';

```

Next, create the capture and apply processes and define the replication rules.

```

CONNECT SYS/SYS_PASSWORD AS SYSDBA

ALTER SYSTEM ARCHIVE LOG CURRENT;

CONNECT strmadmin/strmadminpw

--- -----
--- Stop the capture process if it's already active.
--- -----

BEGIN

```

```

        DBMS_CAPTURE_ADM.STOP_CAPTURE(
            capture_name => 'db2_capture');
END;
/

-----
--- Stop the apply process if it's already active.
-----

BEGIN
    DBMS_APPLY_ADM.STOP_APPLY(
        apply_name => 'apply_2_db2');
END;
/

-----
--- Define the capture rule, this one captures changes to scott.emp
-----

BEGIN
    DBMS_STREAMS_ADM.ADD_SCHEMA_RULES(
        schema_name => 'scott',
        streams_type => 'capture',
        streams_name => 'db2_capture',
        queue_name => 'strmadmin.streams_queue',
        include_dml => true,
        include_ddl => true);
END;
/

-----
--- Set the capture instantiation level
-----

DECLARE
    iscn NUMBER; -- Variable to hold instantiation SCN value
BEGIN
    iscn := DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER();

    DBMS_APPLY_ADM.SET_TABLE_INSTANTIATION_SCN(
        source_object_name => 'scott.emp',
        source_database_name => 'ORAv92',
        instantiation_scn => iscn,
        apply_database_link => 'strmdblink.your.domain.com');
END;
/

-----
--- Drop the apply process if it already exists.
-----

```

```
-----  
BEGIN  
    DBMS_APPLY_ADM.DROP_APPLY(  
        apply_name => 'apply_2_db2');  
END;  
/  
  
-----  
--- Create the apply process  
-----  
BEGIN  
    DBMS_APPLY_ADM.CREATE_APPLY(  
        queue_name => 'strmadmin.streams_queue',  
        apply_name => 'apply_2_db2',  
        apply_database_link => 'strmdblink.your.domain.com',  
        apply_captured => true);  
END;  
/  
  
-----  
--- Create the apply rule  
-----  
BEGIN  
    DBMS_STREAMS_ADM.ADD_TABLE_RULES(  
        table_name => 'scott.emp',  
        streams_type => 'apply',  
        streams_name => 'apply_2_db2',  
        queue_name => 'strmadmin.streams_queue',  
        include_dml => true,  
        include_ddl => false,  
        source_database => 'ORAv92');  
END;  
/  
  
-----  
--- Turn on tracing for the apply process (be careful, this  
--- generates alot of output).  
-----  
BEGIN  
    DBMS_APPLY_ADM.SET_PARAMETER(  
        apply_name => 'apply_2_db2',  
        parameter => 'trace_level',  
        value => 127 );  
END;  
/  
  
-----
```

```
-----  
--- Turn off disable_on_error for the apply process  
-----  
BEGIN  
  DBMS_APPLY_ADM.SET_PARAMETER(  
    apply_name => 'apply_2_db2',  
    parameter => 'disable_on_error',  
    value => 'n');  
END;  
/  
  
-----  
--- Start the apply process.  
-----  
BEGIN  
  DBMS_APPLY_ADM.START_APPLY(  
    apply_name => 'apply_2_db2');  
END;  
/  
  
-----  
--- Start the capture process.  
-----  
BEGIN  
  DBMS_CAPTURE_ADM.START_CAPTURE(  
    capture_name => 'db2_capture');  
END;  
/
```

For more detailed information on Oracle streams replication, refer to the Oracle9i Streams manual.

Copying Data to the Oracle Database Server from the DB2 Server

Use one of the following options to copy data from the DB2 server to the Oracle database server:

- Use the CREATE TABLE command to copy data from the DB2 server to the Oracle database server. To create a table on your local database and insert rows from a DB2 table, use:

```
CREATE TABLE table_name AS query;
```

The next example creates the table EMP in the local Oracle database server and inserts the rows from the EMP table on the DB2 server:

```
CREATE TABLE EMP AS SELECT * FROM SCOTT.EMP@gateway;
```

- Use the INSERT command to copy data from the DB2 server to the Oracle database server:

```
INSERT INTO oracle_table SELECT * FROM db2table@gateway;
```

The following example selects all rows from the EMP table on the DB2 server and inserts them into the local Oracle EMP table:

```
INSERT INTO EMP SELECT * FROM SCOTT.EMP@gateway;
```

- Use the CREATE SNAPSHOT command to automatically and asynchronously copy DB2 server data into the Oracle database server. The complete refresh capability can be used to propagate a complete copy or a subset. For more information on creating SNAPSHOTS, refer to the *Oracle9i Server SQL Reference*. To create a copy:

```
CREATE SNAPSHOT empdb2
  PCTFREE 5 PCTUSED 60
  TABLESPACE users
  STORAGE (INITIAL 50K NEXT 50K)
  REFRESH COMPLETE NEXT SYSDATE + 1
  WITH ROWID
  AS
  SELECT * FROM SCOTT.EMP@gateway;
```

The following example creates a snapshot of data that is refreshed every day after the first refresh. If you only require a subset of the DB2 data, then a WHERE clause is added as in the following example:

```
CREATE SNAPSHOT empdb2
  PCTFREE 5 PCTUSED 60
  TABLESPACE users
  STORAGE (INITIAL 50K NEXT 50K)
  REFRESH COMPLETE NEXT SYSDATE + 1
  WITH ROWID
  AS
  SELECT * FROM SCOTT.EMP@gateway
  WHERE deptno=20;
```

- Use the SQL*Plus COPY command to copy data from the DB2 server to the Oracle database server:

```
COPY FROM username/password@gateway
INSERT destination_table
USING query;
```

The following example selects all rows from the EMP table in DB2 and inserts them into the local Oracle EMP table:

```
COPY FROM SCOTT/TIGER@gateway
INSERT EMP
USING SELECT * FROM SCOTT.EMP@gateway;
```

Note: The Oracle9i database server allows only one LONG column per table, which might create a situation where a DB2 table cannot be replicated directly in an Oracle table.

Developing Applications

Oracle Transparent Gateway for DB2 allows applications written for the Oracle database server to access tables in a DB2 database. Using a database link, the access can be made transparent by using synonyms or views of the DB2 tables. However, there are fundamental SQL, data type, and semantic differences between the Oracle database server and the DB2 database. Read this chapter to learn these differences and for information on developing applications.

The following topics are included:

- [Gateway Appearance to Application Programs](#) on page 9-2
- [Array Processing](#) on page 9-2
- [Using Oracle Stored Procedures with the Gateway](#) on page 9-3
- [Using DB2 Stored Procedures with the Gateway](#) on page 9-4
- [Passing DB2 SQL Statements through the Gateway](#) on page 9-7
- [DB2 Data Types to Oracle Data Type Conversion](#) on page 9-10
- [SQL Functions](#) on page 9-18
- [Oracle Database Server SQL Construct Processing](#) on page 9-19
- [Oracle Database Server and DB2 Differences](#) on page 9-20
- [Oracle Data Dictionary Emulation on a DB2 Server](#) on page 9-21

Gateway Appearance to Application Programs

An application written to access information in a DB2 database interfaces with an Oracle database server. When developing applications, remember the following:

- You must define the DB2 database to the application by use of a database link. Your application specifies tables existing on a DB2 database using the name defined in the database link. For example, if you define a database link naming the DB2 database link DB2, and an application needs to retrieve data from an Oracle database server and the DB2 database, then the following SQL statement retrieves data from both Oracle and DB2:

```
SELECT EMP.EMPNO, EMPS.SALARY FROM EMP, EMPS@DB2
       WHERE EMP.EMPNO = EMPS.EMPNO;
```

In this example EMP, is a table on the Oracle database server and EMPS is a table on the DB2 server. Alternatively, you can define a synonym or a view on the DB2 server table and access the information without the database link suffix.

- You can perform reads and writes of data to a defined DB2 database. SELECT, INSERT, UPDATE, and DELETE are all valid operations.
- A single transaction can write to one DB2 database and to multiple Oracle database servers.
- Single SQL statements, using a JOIN, can refer to tables in multiple Oracle database servers, multiple DB2 databases, or both.

Array Processing

When evaluating and tuning your gateway configuration, you can achieve performance gains by using the Oracle array processing interface. An array is a collection of data items, called elements, associated with a single variable. With arrays, you can use a single SQL statement to manipulate an entire collection of data items. For example, suppose you want to insert information regarding 100 employees into the EMP table on DB2. Without arrays, your program must do 100 individual INSERTs—one for each employee. With arrays, only one INSERT is necessary.

The use of array processing reduces network calls, which can save elapsed time and CPU cycles. In addition, when using INSERT for multiple rows, DB2 processing is optimized by retaining the original SQL statement for repeated running.

You can set the array size between the client and the gateway by using your Oracle application implementation for UPDATE, DELETE, and INSERT.

For more information on array processing usage and implementation in your Oracle application, refer to the Oracle *SQL*Plus User's Guide and Reference* manual or *Programmer's Guide to the Oracle Call Interface*.

Note: For performance reasons, Oracle Corporation recommends setting the initial Oracle application array size between 10 and 100.

Fetch Reblocking

The Oracle database server supports fetch reblocking with the HS_RPC_FETCH_REBLOCKING parameter.

When the value of this parameter is set to ON (the default), the array size for SELECT statements is determined by the HS_RPC_FETCH_SIZE value. The HS_RPC_FETCH_SIZE parameter defines the number of bytes sent with each buffer from the gateway to the Oracle database server. The buffer might contain one or more qualified rows from DB2. This feature can provide significant performance enhancements, depending on your application design, installation type, and workload.

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

Refer to "[Step 10: Gateway Service Definition](#)" on page 5-29 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 a set of SQL and other PL/SQL programming language statements together to perform a specific task. Oracle stored procedures are stored in the database for continued use. Applications use standard Oracle PL/SQL to call stored procedures.

Oracle stored procedures can be located in a local instance of the Oracle database server and a remote instance. The following example shows 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.

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.

`Empproc` is an Oracle stored procedure, which subsequently accesses data in DB2 using the gateway:

Like the Oracle database server, standard PL/SQL is used to create and run the procedure. There is no difference with the gateways, except the stored procedure is accessing DB2 instead of the Oracle database server.

Gateway two-phase commit processing also applies to updates to DB2 being made within an Oracle stored procedure. This means the stored procedure can update a single instance of DB2 while also updating any number of Oracle database servers within a single transaction.

Using DB2 Stored Procedures with the Gateway

The procedural feature of the gateway enables completion of native DB2 stored procedures. In other words, the stored procedure is no longer defined in the Oracle database server, but instead, is defined to DB2. Again, standard Oracle PL/SQL is used by the Oracle application to run the DB2 stored procedure.

The gateway does not require special definitions to call the DB2 stored procedure. Once the stored procedure is defined to DB2, the gateway is able to use the existing DB2 definition to run the procedure.

An Oracle application calls the `empproc` stored procedure that is defined to DB2.

From the application's perspective, running the DB2 stored procedure is no different than invoking a stored procedure at a remote Oracle instance.

Oracle Application DB2 Stored Procedure Completion

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 using the procedures found in the IBM documents for your platform and operating system.

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:

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

When the gateway receives a call to run a DB2 stored procedure, it first does a lookup of the procedure in the SYSIBM.SYSPROCEDURES DB2 catalog table to determine:

- The stored procedure to be run.

The gateway retrieves information from the PROCEDURE, AUTHID, and LUNAME columns to locate the DB2 stored procedure to be called by the gateway. Multiple DB2 stored procedures can have the same PROCEDURE name, but must be uniquely identified by the combination of PROCEDURE, AUTHID, and LUNAME columns with the DB2 SYSPROCEDURES catalog table.

When the gateway is performing a lookup in the SYSPROCEDURES DB2 catalog table, it uses the userid passed from the database link or a blank for determining the proper authid for the DB2 stored procedure. When searching for the LUNAME, the gateway uses the DB2 CURRENT SERVER value or a blank for determining the proper LUNAME for the DB2 stored procedure. In general, the gateway attempts to find a non blank field in AUTHID and LUNAME first when attempting to locate the proper DB2 stored procedure to run.

An example of the order the gateway would use to choose a DB2 PROC1 stored procedure that has different AUTHID and LUNAME entries is:

PROCEDURE	AUTHID	LUNAME
PROC1	USERID1	DSN
PROC1	USERID1	<blank>

PROCEDURE	AUTHID	LUNAME
PROC1	<blank>	DSN
PROC1	<blank>	<blank>

- The parameter list of the stored procedure.

The gateway uses the PARMLIST field in SYSPROCEDURES to determine what parameter list the stored procedure expects. The gateway automatically converts DB2 data types to and from PL/SQL. For gateway data type conversion rules, refer to "[DB2 Data Types to Oracle Data Type Conversion](#)" on page 9-10.

When running a DB2 stored procedure, a two-part procedure name is sent to DB2 through the gateway. If no qualifier is used in the Oracle application to call the stored procedure, then the userid passed over the database link, or PUBLIC, is used as the qualifier for the procedure name.

DB2 stored procedures for DB2 V5.1 require that SYSPROC must be the first qualifier of a stored procedure name. Therefore, the application must ensure SYSPROC is used as the qualifier for the DB2 stored procedure. One way to do this is to explicitly qualify the procedure name:

```
BEGIN
  SYSPROC.PROC1 (parm1)
END
```

DB2 stored procedures for DB2 V6.1 and subsequent allows stored procedures to be qualified by schema name other than SYSPROC.

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 if you are running a DB2 stored procedure from an Oracle application, and 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 rollback 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.

- When running DB2 stored procedures containing DB2 SQL, you must have the collection ID of the DB2 package specified in the SYSIBM.SYSPROCEDURES or SYSROUTINES DB2 data dictionary table for the DB2 stored procedure that is run.

This is required because the DB2 plan for the gateway must identify the packages for the DB2 stored procedures it runs. The default DB2 bind JCL, delivered with the gateway, uses * . * to identify the package list for the gateway. When this is specified, DB2 identifies the collection ID of the DB2 stored procedure as the one specified in the COLLID column of the DB2 stored procedure entry in the SYSIBM.SYSPROCEDURES or SYSROUTINES table.

- PL/SQL records cannot be passed as parameters when invoking a DB2 stored procedure.
- The gateway supports both SIMPLE and SIMPLE WITH NULLS linkage conventions of DB2 stored procedures.
 - The SIMPLE linkage convention means that the parameters passed to and from the DB2 stored procedure cannot be null.
 - The SIMPLE WITH NULLS linkage convention means that the parameters passed to and from the DB2 stored procedure can be null when they are passed using indicator variables.
 - Embedded PL/SQL or OCI can be used in the host program to operate on indicator variables. Refer to [Appendix G, "Sample Applications"](#), for a sample DB2 stored procedure and PL/SQL program using the SIMPLE WITH NULLS linkage convention.

Passing DB2 SQL Statements through the Gateway

The passthrough SQL feature allows an application developer to send a SQL statement directly to DB2 without the statement being interpreted by the Oracle database server. The DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE SQL passthrough statements supported by the gateway are limited to non queries (INSERT, UPDATE, DELETE, and DDL statements) and cannot contain bind

variables. The gateway can run native DB2 SQL statements using `DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE`.

`DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE` is a built-in gateway function. This function receives one input argument and returns the number of rows affected by the SQL statement. For DDL statements, the function returns zero.

`DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE` are reserved names of the gateway and are used specifically for running native DB2 SQL.

This release of Oracle Transparent Gateway for DB2 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 9-9 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:

```
number_of_rows = DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink ('native_DB2_
sql');
where:
```

<code>number_of_rows</code>	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.
<code>dblink</code>	is the name of the database link used to access the gateway.
<code>native_DB2_sql</code>	is a valid DB2 non query SQL statement (except <code>CONNECT</code> , <code>COMMIT</code> , and <code>ROLLBACK</code>). The statement cannot contain bind variables. DB2 SQL statements that cannot be dynamically prepared are rejected by DB2. The SQL statement passed by the <code>DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE</code> function must be a character string. For more information regarding the DB2 SQL statements, refer to the IBM documents for your platform and operating system.

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;
/
    
```

1. 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 DB2 provides a facility to retrieve results sets from a SELECT SQL statement entered through passthrough. Refer to *Oracle9i Server Distributed Database Systems* for additional information.

Example

```

DECLARE
    CRS binary_integer;
    RET binary_integer;
    VAL VARCHAR2(10)
BEGIN
    CRS:=DBMS_HS_PASSTHROUGH.OPEN_CURSOR@gtwlink;
    DBMS_HS_PASSTHROUGH.PARSE@gtwlink(CRS,'SELECT NAME FROM PT_TABLE');
BEGIN
    RET:=0;
    WHILE (TRUE)
    LOOP
        RET:=DBMS_HS_PASSTHROUGH.FETCH_ROW@gtwlink (CRS,FALSE);
        DBMS_HS_PASSTHROUGH.GET_VALUES@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;
/

```

DB2 Data Types to Oracle Data Type Conversion

To move data between applications and the underlying database, the gateway maps data values from a host variable or literal of a specific data type to a data type understood by the underlying database.

Oracle tools expect Oracle data types. Consequently, the gateway maps values from DB2 servers into appropriate Oracle data types before passing these values back to the application or Oracle tool. The data type mapping and restrictions are:

DB2 Server	Oracle External	Criteria
CHAR(N)	CHAR(N)	N=<254*
VARCHAR(N)	VARCHAR2(N) LONG	N=<4000 (refer to Note 1) 4000<N<DB2 maximum long value
LONG VARCHAR(N)	VARCHAR2(N) LONG	N=<4000 (refer to Note 1) 4000<N<DB2 maximum long value
CHAR(N) FOR BIT DATA	RAW(N)	N=<254 (refer to Note 2)
VARCHAR(N) FOR BIT DATA	RAW(N) LONG RAW(N)	1=<N=<255 (refer to Note 1) 255<N=<DB2 maximum long value
LONG VARCHAR(N) FOR BIT DATA	RAW(N) LONG RAW(N)	1=<N=<255 (refer to Note 1) 255<N=<DB2 maximum long value
DATE TIME TIMESTAMP	DATE CHAR(8) CHAR(26)	Refer to " Performing Date and Time Operations " on page 9-12
GRAPHIC (N)	CHAR(2N+2)	N<126

DB2 Server	Oracle External	Criteria
VARGRAPHIC (N)	VARCHAR(2N+2)	N<1999
LONG VARGRAPHIC (N)	VARCHAR2(2N+2)	
FLOAT(N) (single)	FLOAT(21)	1=<N=<21
FLOAT(N) (double)	FLOAT(53)	22=<N=<53
Decimal(P,S)	NUMBER(P,S)	n/a
INTEGER	NUMBER(10)	n/a
SMALLINT	NUMBER(5)	n/a

Note 1: Although the limits of some data types within the Oracle database server has increased, the limits used in the gateway are not changed. This is so that you can maintain existing user application compatibility.

Note 2: The Oracle database server can support a length of up to 2000 fixed character columns; but the maximum for DB2 is 254.

Note: The Oracle9i database server allows only one LONG column per table. This might allow for a situation where a DB2 table cannot be directly replicated as an Oracle table.

To determine DB2 maximum long values, refer to the IBM documents for your platform and operating system.

Performing Character String Operations

Frequently, DB2 databases are designed to hold non-character binary data in character columns. Applications run on DB2 systems can store and retrieve data as though it contained character data. However, when an application accessing this data runs in an environment using a different character set, inaccurate data might be returned.

When character data is sent to DB2 from an ASCII system, ASCII data is translated to EBCDIC. This translation is meaningless when the characters are binary data in a character column. The application receives incorrect information or errors.

To resolve these errors, the gateway requires character columns on DB2 holding non-character data be created with the FOR BIT DATA option. In the application, the character columns holding non-character data can be processed using the Oracle

data types 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.

Existing DB2 tables can be changed by directly updating the DB2 catalog.

For more information about DB2 parameters, refer to the IBM documents for your platform and operating system.

Converting Character String Data Types

The DB2 VARCHAR data type can be from one to the maximum long value for DB2. This data type is converted to an Oracle VARCHAR2 data type if it is between 1 and 4000 characters in length. If character length is between 4000 and the maximum long value for DB2, then it is converted to an Oracle LONG data type.

For additional information about determining the maximum long value for DB2, refer to the IBM documents for your platform and operating system.

The Oracle LONG data type can be from 1 to 2G in length, but the DB2 VARCHAR data type can be no longer than 32 740 bytes. If you define a LONG data type longer than 32 740 bytes in length, then you receive an error message.

Performing Date and Time Operations

The implementation of date and time data differs significantly in DB2 databases and the Oracle database server. The Oracle database server has a single date data type, DATE, containing both calendar date and time of day information. DB2 databases support the following three distinct date and time data types:

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 of the time value.

There is no built-in mechanism that translates the IBM TIME and TIMESTAMP data to Oracle DATE data. An application must process TIME data types in the Oracle CHAR format with a length of eight bytes. An application must process the TIMESTAMP data type in the Oracle CHAR format with a length of 26 bytes.

An application reads TIME and TIMESTAMP columns as character strings and converts or subsets portions of the string to perform numerical operations. TIME

and `TIMESTAMP` values can be sent to a DB2 database as character literals or bind variables of the appropriate length and format.

Oracle and IBM `DATE` data types 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 the Oracle database server format and DB2 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 a DB2 compatible format.

DB2 Local Date Exit

Oracle Transparent Gateway for DB2 includes a DB2 local date exit. The exit is called only when needed and does not interfere with normal DB2 operations or impact performance. With the exit installed, DB2 `DATE` columns are handled through the gateway. If you do not install the exit, then Oracle SQL requires changes when referencing DB2 `DATE` columns.

When a string constant, string bind variable, string expression, or character column is compared or assigned to a DB2 date column, it is converted from its string format to an internal DB2 format before DB2 processes it. DB2 date conversion routines look for the following formats of date string. The DB2 local date exit is called only if the date string does not match any of the standard formats.

DB2 Date Format	Pattern	Example
EUR	DD.MM.YYYY	30.10.1994
ISO	YYYY-MM-DD	1994-10-30
JIS	YYYY-MM-DD	1994-10-30
LOCAL	DD-MON-YY	30-OCT-94
	DD-MON-RR	29-MAR-05
USA	MM/DD/YYYY	10/30/1994

The LOCAL DB2 date format is available when the gateway local date exit is installed.

Any non-gateway DB2 access (for example, through batch, TSO, or CICS) supplying date strings (ISO, JIS, USA, or EUR formats) do not call the gateway's DB2 local date exit. The local exit is called only if the date format cannot be matched to ISO, JIS, USA, or EUR formats. In a native DB2 program, this is frequently due to a bad date string value. If a bad date string value is entered, then the DB2 local date exit is called and rejects the bad date string.

The DB2 local date exit is called in the following circumstances:

- A native DB2 program has a bad date string value that cannot be matched to ISO, JIS, USA, or EUR formats.
- A gateway program supplies one of the Oracle DATE formats, 'DD-MON-YY' or 'DD-MON-RR'.

When you install the DB2 local date exit (DSNXVDTX) supplied with the gateway, you can use ISO, JIS, USA, and EUR as well as the Oracle date formats 'DD-MON-YY', 'DD-MON-YYYY', and 'DD-MON-RR' through the gateway. The DB2 local date exit must be installed in order to specify these Oracle date formats without any SQL changes. If you do not install the exit, then you must use the Oracle TO_DATE function to pass these Oracle date formats through the gateway. Refer to ["Step 4: Make Authorization and Local Date Exits Available to DB2"](#) on page 5-11 for further information.

Date Considerations in SQL Coding

If the gateway local date exit is installed on the DB2 system, then DB2 DATE columns appear as Oracle DATE columns through the gateway. Normal Oracle DATE processing and string values can be used. DB2 DATE columns are handled as Oracle DATE columns.

If the gateway's DB2 local date exit is not installed, then most SQL statements referencing DB2 DATE columns require changes. When a string constant or string bind variable is compared or assigned to a DB2 DATE column, a TO_DATE function must be added to the statement, enclosing the constant or bind variable.

When the DB2 local date exit is installed, the following SQL statements are accepted:

```
INSERT INTO EMP (HIREDATE) VALUES ('30-OCT-94');
SELECT * FROM EMP WHERE HIREDATE = '30-OCT-94';
UPDATE EMP SET HIREDATE = '31-OCT-94'
WHERE HIREDATE = '30-OCT-94';
DELETE FROM EMP WHERE HIREDATE = '31-OCT-94';
```

If the DB2 local date exit is not installed, then the following SQL statements are required:

```
INSERT INTO EMP (HIREDATE) VALUES (TO_DATE('30-OCT-94'));
SELECT * FROM EMP WHERE HIREDATE = TO_DATE('30-OCT-94');
UPDATE EMP SET HIREDATE = TO_DATE('31-OCT-94')
    WHERE HIREDATE = TO_DATE('30-OCT-94');
DELETE FROM EMP WHERE HIREDATE = TO_DATE('31-OCT-94');
```

NLS_DATE_FORMAT Support

The following patterns can be used for the NLS_DATE_FORMAT:

DB2 Date Format	Pattern	Example
EUR	DD.MM.YYYY	30.10.1994
ISO	YYYY-MM-DD	1994-10-30
JIS	YYYY-MM-DD	1994-10-30
USA	MM/DD/YYYY	10/30/1994

The Oracle default format of 'DD-MON-YY' is not allowed with DB2. As a result, the gateway local date exit is provided to change the Oracle default date format of 'DD-MON-YY' or 'DD-MON-RR' to the DB2 ISO format of 'YYYY-MM-DD' before passing the date to DB2.

The following example demonstrates the most efficient way to enter and select date values in the twenty-first century:

```
ALTER SESSION SET 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';
```

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.

Date Arithmetic

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

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

The date and number addition and subtraction (*date + number*, *number + date*, *date - number*) forms are sent through to the DB2 server where they are rejected. The supported servers do not allow number addition or subtraction with dates. Because of differing interpretations of date subtraction in the supported servers, subtracting two dates (*date1 - date2*) does not work correctly when post processed by the integrating server.

Note: Oracle Corporation recommends avoiding date arithmetic expressions in all gateway SQL statements.

Performing Numeric Data Type Operations

DB2 servers perform automatic conversions to the numeric data type of the destination column (such as integer, double-precision floating point, or decimal). You have no control over the data type conversion, and this conversion might be independent of the data type of the destination column in the database.

For example, if PRICE is an integer column of the PRODUCT table in a DB2 database, then the update shown in the following example inaccurately sets the

price of an ice cream cone to \$1.00 because the DB2 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 DB2 server automatically converts the decimal data value of 1.50 to 1.

Oracle ROWID Column

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

- UPDATE and DELETE are not supported with the WHERE CURRENT OF CURSOR clause.
- When these statements are used in precompiler and PL/SQL programs, they rely internally on the Oracle ROWID function.
- Oracle fast refresh snapshots between the Oracle database server and DB2 are not supported

Oracle fast refresh snapshots rely internally on the Oracle ROWID column. However, complete refresh snapshots are supported.

Double Byte Character Set Support

DB2 CHAR and VARCHAR for bit data (RAW data types) are supported. Raw data in VALUES clause, WHERE clause predicate, or bind variables are treated as hexadecimal digits.

Katakana is not supported in the DB2 GRAPHIC data type because the data type is double byte only. Katakana is encoded as single byte in IBM code pages 290 and 1027, and Oracle JA16DBCS and JA16EBCDIC930 character sets.

Katakana can be supported in DB2 CHAR and VARCHAR data types as mixed data if Oracle client programs and the Oracle database server are linked with NLSRTL release 2.3.4 or later. This uses the correct Katakana translation routines.

CHAR FOR BIT DATA

CHAR FOR BIT DATA is fixed length binary data in DB2. In the Oracle database server, CHAR FOR BIT DATA is converted to RAW, which is in variable length binary format.

Ensure your programs:

- Handle CHAR FOR BIT DATA as a variable length string of binary characters.
- Set the length to the maximum length of the DB2 fixed length column and pad with the binary value your program expects to be returned.

SQL Functions

One of the most important features of the Oracle Open Gateways product family is providing SQL transparency to the user and the application programmer. Foreign data store SQL functions can be categorized into three areas:

- **Compatible**

SQL functions with the same meaning and results on both the Oracle database server and foreign data stores. Some examples of compatible SQL functions include:

- AVG
- CONCAT
- COUNT (*) only
- COUNT(DISTINCT *expression*)
- MAX
- MIN
- SUM

- **Translated**

SQL functions that provide the same functionality, but are referenced by a different name, at the Oracle database server and the foreign data store. Translated SQL functions include:

Oracle Database Server Functions	DB2 Functions
NVL	VALUE

- **Compensated**

Advanced SQL functions that are supported by the Oracle database server and that cannot be expressed or recognized by the foreign data store.

SQL compensation in the Oracle Open Gateways enriches the semantics of the native SQL of a remote data source, such as DB2. This important feature of the gateway allows application developers and users to leverage the advanced features of the Oracle database server.

Refer to [Appendix F, "Quick Reference to Oracle SQL Functions"](#), for a listing of the Oracle database server functions. For more detailed information, refer to the *Oracle9i Server SQL Reference*.

Oracle Database Server SQL Construct Processing

Some gateway post-processing considerations are explained below.

SELECT without the FOR UPDATE Clause

A SELECT without the FOR UPDATE clause can be handled in one of three ways:

- If the entire WHERE clause of the SELECT statement is acceptable syntax for DB2, then it is given to DB2 to perform.
- If part, but not all, of the WHERE clause of the SELECT statement uses features not available in DB2, then the WHERE clause is split between the DB2 system and the Oracle database server.

The portion of the WHERE clause acceptable for DB2 is sent to DB2. The Oracle database server post processes the results of the DB2 SELECT and applies the Oracle-specific WHERE clause elements. This results in DB2 doing as much of the WHERE clause as possible.

- If the entire WHERE clause is not acceptable for DB2, then an unqualified SELECT (without the WHERE clause) is sent for DB2, and the Oracle database server post processes the entire WHERE clause.

The Oracle database server post processes SELECT statements without the FOR UPDATE clause. Most Oracle SELECT statements are supported. One exception is the CONNECT BY clause.

SELECT FOR UPDATE, INSERT, and DELETE Clauses

DB2 must process the entire SELECT FOR UPDATE, INSERT, and DELETE clauses. The Oracle database server cannot post process these clauses. Only SQL that is a common subset of the Oracle database server and DB2 SQL can be used with these statements.

The following rules exist for the use of SELECT FOR UPDATE, INSERT, and DELETE clauses:

- Only Oracle syntax that is also valid for DB2 can be used. For DB2 SQL syntax, refer to the IBM documents for your platform and operating system.
- The following Oracle database server functions are supported with all options:
 - AVG
 - MAX
 - MIN
 - SUM
 - TO_DATE
- The NOWAIT option of the FOR UPDATE clause of the SELECT statement is not supported.
- Although DB2 requires a list of column names in the FOR UPDATE clause, the unqualified Oracle syntax FOR UPDATE with no column names is accepted. The gateway derives the column list from the SELECT result column list.

Oracle Database Server and DB2 Differences

Please be aware of the following differences between the Oracle Database Server and DB2.

Mass Delete from a Segmented Tablespace

When using the following command from SQL*Plus:

```
DELETE FROM ABC@dblink
```

all rows are deleted from a segmented tablespace. However, DB2 occasionally sets the updated rows field to negative 1 (-1) even though more rows are actually deleted. This can cause the result from SQL*Plus to indicate an incorrect number of rows updated.

Mapping the COUNT Function

The Oracle database server supports four options for the COUNT function:

- COUNT(*)
- COUNT(DISTINCT expression)
- COUNT(ALL expression)
- COUNT(expression)

DB2 servers support only two options for the COUNT function:

- COUNT(*)
- COUNT(DISTINCT expression)

COUNT(ALL expression) and COUNT(expression) are post processed.

Oracle Bind Variables

Oracle bind variables become DB2 parameter markers when used with the gateway. Therefore, the bind variables are subject to the same restrictions as DB2 parameter markers. For example, the following statements are not allowed:

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

For more information about DB2 parameter marker restrictions, refer to the IBM documents for your platform and operating system.

Oracle Data Dictionary Emulation on a DB2 Server

The gateway can optionally augment the DB2 database catalogs with data dictionary views modeled after the Oracle data dictionary. These views are based upon the dictionary tables in the DB2 database, presenting the 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 the gateway users with an interface (that looks like an Oracle database server interface) to the contents and use of the DB2 database. Some of these views are required by Oracle products.

You can query the gateway data dictionary views to look at the objects in the DB2 database and to determine the authorized users of the DB2 database.

All Oracle DB2 catalog views are supported in this release of the gateway. Refer to [Appendix E, "Data Dictionary Views"](#), for descriptions of Oracle DB2 catalog views.

DB2 Special Registers

You are able to access DB2 special registers using the gateway. During installation of the gateway, a DB2 view is created to access special registers. For example, in order to find out the primary authorization ID being used by the gateway, run the following statement from your application:

```
SELECT CURRENT_USER FROM OTGDB2.OTGREGISTER@DB2
```

where OTGDB2 is the default qualifier of the OTGREGISTER view, and DB2 is the name of a database link to the gateway. Refer to [Appendix E, "Data Dictionary Views"](#), for a description of the OTGREGISTER view.

Error Messages, Diagnosis, and Reporting

This chapter discusses error messages generated by Oracle Transparent Gateway for DB2, the diagnosis of suspected Oracle database server errors, and the requirements for documenting these errors to Oracle Support Services.

For information on Oracle9i Enterprise Edition for OS/390-specific error messages, refer to the *Oracle9i Enterprise Edition Messages Guide for OS/390*.

The following topics are included:

- [Message and Error Code Processing](#) on page 10-2
- [Oracle Support Services](#) on page 10-5
- [Providing Error Documentation](#) on page 10-5
- [General Documentation Requirements](#) on page 10-6
- [Error Diagnosis](#) on page 10-7
- [Error Categories](#) on page 10-7
- [System Dumps](#) on page 10-10
- [GTF](#) on page 10-11

Message and Error Code Processing

The gateway architecture includes a number of separate components. Any of these components can detect and report an error condition while processing a SQL statement referring to one or more DB2 database tables. An error condition can be complex, involving error codes and supporting data from multiple components. In all cases, the application ultimately receives a single Oracle database server error code upon which to act.

Error conditions are represented in one of two ways:

- Mapped

When possible, an error code from the DB2 database is converted to the Oracle database server error code associated with the same logical condition.

Error code mapping is provided to support application designs that test for and act upon specific error conditions. The set of mapped errors is limited to those associated with conditions common to most relational databases.

- Messages from the gateway

Most gateway error conditions are reported to the application using one of the gateway error codes in the range of ORA-9100 through ORA-9199. These messages are less closely linked to specific DB2 database conditions. The message format is explained in "[Interpreting Message Formats](#)" on page 10-3.

The ORA-9100 error code is returned for all errors for which a more specific error code does not exist. When an ORA-9100 error code is returned, the error might have been caused in the gateway by a DB2 support component on the target database system.

Mapping DB2 Error Messages to Oracle Error Messages

DB2 error messages, that is, SQLSTATE codes, are mapped to Oracle database server error codes. Notice that multiple DB2 SQLCODE can refer to the same Oracle database server error code.

Description	SQLSTATE Code	Oracle Database Server Error Code
No rows selected	0200	0
Unique index constraint violated	23505	ORA-0001

Description	SQLSTATE Code	Oracle Database Server Error Code
Table or view does not exist	52004 or 42704	ORA-00942
Object name greater than 18 characters and, therefore, object does not exist	54003 or 42622	ORA-00942
Insufficient privileges	42501	ORA-01031
Divisor is equal to zero	01519 or 01564	ORA-01476

Interpreting Message Formats

Error messages are generally accompanied by additional message text, beyond the text associated with the Oracle database server message number. The additional text includes details about the error.

Most gateway messages exceed the 70 character message area in the Oracle SQLCA. Use SQLGLM or OERHMS in the programmatic interfaces and the OCI that you use with the gateway to view the entire message. Refer to the *Programmer's Guide to the Oracle Precompilers* for information about SQLGLM and the *Programmer's Guide to the Oracle Call Interface* for information about OERHMS.

Gateway messages use the following format:

```
ORA-nnnn:error message text
gateway message line(s)
ORA-2063:PRECEDING n LINES FROM dblink
```

where:

- nnnn* is an Oracle database server error number. If *nnnn* is between 9100 and 9199, then the message is from the gateway. If it is not in this range, then it is a mapped error message.
- error message text is the text of the message associated with the error.
- gateway message line(s) are additional messages generated by the gateway. The gateway messages lines are described in ["Diagnosing Errors Detected by the Oracle Database Server"](#) on page 10-5.

`n` is the total number of gateway message lines.
`dblink` is the name of the database link that is used to access the gateway.

Messages Generated by Oracle Transparent Gateway for DB2

The following message is generated by Oracle Transparent Gateway for DB2.
ORA-02063 preceding `n` lines from `dblink`

Examples

ORA-02063 PRECEDING 3 LINES FROM GTWLINK

Cause: Indicates an error from Oracle Transparent Gateway for DB2. The `dblink` in the ORA-2063 message indicates the name of the database link that was used to access the gateway.

Action: All error messages need actions.

ORA-28500 CONNECTION FROM ORACLE TO NON-ORACLE SYSTEM RETURNED THIS MESSAGE:

DSNT408I SQLCODE = -084, ERROR: UNACCEPTABLE SQL STATEMENT
DSNT418I SQLSTATE = 371512 SQLSTATE RETURN CODE
DSNT415I SQLERRP=DSNHAPLY SQL PROCEDURE DETECTING ERROR

Cause: Indicates an error from DB2. It is followed by messages from the DB2 database. The `n` in the ORA-02063 message indicates the total number of gateway message lines referenced in the ORA-28500 message.

Action: All error messages need actions.

ORA-09611 DEFINITION OF TARGET SYSTEM DATA OBJECT IS INCONSISTENT

Cause: If this message is returned while attempting to run a DB2 stored procedure, then it is an indication that the definition of the DB2 stored procedure parameter list is inconsistent with the parameter list passed from the Oracle application.

Action: All error messages need actions.

For additional information on the DB2 messages that are included in the ORA-28500 message, refer to the IBM documents for your platform and operating system.

Diagnosing Errors Detected by the Oracle Database Server

If an error is detected by the Oracle database server, then the gateway message lines do not occur. For example, if the gateway cannot be accessed because of an Oracle Net or gateway installation problem, then the gateway message line is not present in the error message.

Another example of error messages without gateway message lines occurs when an INSERT statement attempts to insert data into a table, but does not include values for all of the columns in the table. The following SQL statement causes an error message:

```
SQL> insert into EMP@DB2 values(9999);  
ERROR at line 1:  
ORA-00947: not enough values
```

The ORA-00947 message is not accompanied by gateway message lines because the error is detected by the Oracle database server. The Oracle database server obtains a description of the DB2 table before sending the INSERT statement to the gateway for processing. This allows the Oracle database server to detect when the INSERT statement is invalid.

Oracle Support Services

Oracle Support Services serves as the interface to the Oracle user community. Refer to the applicable Oracle Support Services publications for a discussion of policies and procedures for using their services.

Providing Error Documentation

During the error resolution cycle, Oracle Support Services might request that you provide them with computer readable data. Send computer readable data, not formatted or printed data. The preferred method for providing error documentation is anonymous FTP. Please contact Oracle Support Services for instructions on how and where to provide doc.

If you are requested to send data to the Support Center, then follow the documentation requirements provided in "[General Documentation Requirements](#)" on page 10-6. Failure to follow these requirements might result in the inability to process your tape. This might delay the resolution of any errors you are reporting.

General Documentation Requirements

When you report a suspected error, you might be asked to describe the Oracle subsystem and OS/390 operating system environments in detail. Provide the full version number of each component that has an error. The full version number includes important PUT levels for your OS/390 system.

Before you contact Oracle Support Services, ensure the following information is available:

- Oracle library naming conventions
- Method of accessing the Oracle utilities
- Oracle subsystem name
- Full version of the Oracle gateway
- Full version of the Oracle database server client tools
- Full version of the Oracle utility
- Full version of the third party tool (if applicable)
- PUT level
- RMID of any relevant OS module

In addition to describing the Oracle operational environment, detailed documentation specific to the error might be required. This might include:

- Gateway PARMLIB members
- Console logs and gateway job logs
- Utility SYSOUT
- System diagnostic messages
- Oracle database server error messages
- System dumps
- Gateway trace data sets
- Database engine trace data sets
- Oracle Net trace data sets
- Output from the CLIST PGMDESCC
- Network level trace data sets for TCP/IP or APPC/LU6.2

Keep in mind that more than one error is often associated with a single failure. Describe all errors for the failure being reported. If your application uses Pro*C, Pro*COBOL, or another Oracle Precompiler, then ensure your application displays or prints out all errors it encounters. Without this information, diagnosing the problem is more difficult.

Error Diagnosis

When investigating a potential Oracle gateway error, start by determining which component is failing, where it is failing, and the error category.

Components

When reporting a problem to Oracle Support Services, identify the component suspected of failure, along with its full version and correct release level.

Error Categories

Use the following error categories to describe the error:

- Documentation errors
- Incorrect output
- Oracle database server external error
- Abend
- Program loop
- Performance
- Missing functionality

Documentation Errors

When reporting documentation errors, you are asked to provide the following information:

- Document name
- Document part number
- Date of publication
- Page number

Describe the error in detail. Documentation errors can include erroneous documentation and omission of required information.

Incorrect Output

In general, an incorrect output error exists whenever an Oracle utility produces a result that differs from written Oracle documentation. When describing errors of incorrect output, you need to describe, in detail, the operation of the function in error. Be prepared to describe your understanding of the proper function, the specific Oracle documentation that describes the proper operation of the function, and a detailed description of the incorrect operation.

If you think you have found a software bug, then be prepared to answer the following questions:

- Does the problem occur in more than one Oracle tool? (Examples of Oracle tools are SQL*Plus and Oracle Developer forms.)
- What are the exact SQL statements used to reproduce the problem?
- What are the full version numbers of the Oracle database server, Oracle gateway, and related Oracle software?
- What is the problem and how is it reproduced?

Oracle External Error

Oracle database server error messages are produced whenever an Oracle gateway, server, tool, or DB2 system detects an error condition. Depending on the circumstances, error messages might be fatal or nonfatal to the utility or server.

Be prepared to identify the exact error message and message number received and the complete circumstances surrounding the error.

Abend

Any program check in an Oracle utility or the Oracle gateway address is considered an error. A full system dump is required as documentation if there is a program check.

Ensure the system dump contains all of the private area of the Oracle gateway address space; without it, diagnosis is sometimes impossible.

System abends might or might not indicate a failure of the Oracle subsystem depending upon circumstances. The following abends are not considered Oracle database server failures:

- 013 - open failure
- 122 - canceled by operator
- 222 - canceled by operator
- 322 - CPU time exceeded
- 722 - SYSOUT lines exceeded

Program Loop

A program loop is evident when the Oracle gateway task consumes CPU time rapidly, but no actual work is performed.

Any program loop occurring within an Oracle gateway address space is considered an Oracle gateway failure. Loop conditions are rarely experienced and are considered serious errors. The initial diagnostic approach with a loop consists of a system dump. If a task is in a program loop, then ensure the system dump includes all of the private area of the gateway address space.

Further diagnosis might be required using OS/390 SLIP commands. Oracle Support Services provides specific instructions on the use of SLIP, depending on the circumstances.

Performance

Oracle system performance is determined by many factors, most of which are not within the control of Oracle Corporation. Considerations such as system load, I/O topology, network topology utilization, and DB2 resource availability and utilization, make the documentation of performance errors difficult.

Provide detailed information about the state of your environment when reporting an error. Specific documentation might include:

- CPU type and memory configuration
- Database topology
- I/O topology
- Network topology
- System workload by type

- Oracle database server workload characterization
- Query completion plans
- DB2 threads and resource information

Missing Functionality

Enhancement requests can be opened with Oracle Support Services to request the inclusion of functions and features not currently available with Oracle products. When opening an enhancement request, describe the specific feature or function to be added to the product, and provide a business case to justify the enhancement.

System Dumps

When providing documentation on suspected Oracle database server failures, it might be necessary for you to provide a system dump of the Oracle gateway or utility address spaces. Dumps are initiated through the OS/390 operator interface using the DUMP and SLIP commands, or automatically by the Oracle gateway if it detects a problem.

Dumps sent to Oracle Support Services as documentation for suspected errors must not be formatted. Formatted dumps cannot be used. Formatting a system dump results in a significant delay in processing reported errors, and you might have to send a new, unformatted dump.

When specifying dump parameters in response to an OS/390 DUMP COMM=(' ') command, you must include the following specification:

CSA, PSA, TRT, RGN

System Dump Data Sets

Once a SYS1.DUMPxx data set is created, the system operator is notified whenever a dump to that data set occurs. Because all Oracle abends are dumped to SYS1.DUMP data sets and are not dynamically allocated by OSDI, you must ensure a SYS1.DUMP data set is always available.

You must also ensure the SYS1.DUMP data set is large enough to accommodate the gateway address space.

If a SYS1.DUMP data set is not available, then a dump might be lost.

Operator Initiated Dumps

Operator initiated dumps are accomplished with the OS/390 DUMP command:

```
DUMP COMM=(text)
```

where `text` is the title you want the dump to have.

After the DUMP command has been entered, you must respond to the system WTOR with:

```
R xx, [JOBNAME=(ssn) |ASID=(nnn) , ]SDATA=(CSA,PSA,TRT,RGN)
```

where:

`xx` is the reply identification number.

`ssn` is the name of the Oracle subsystem.

`nnn` is the hexadecimal address space identifier of the address space you want to dump.

GTF

You might need to use GTF as a diagnostic tool under certain circumstances. Oracle Support Services provides specific instructions if this is necessary.

Migration and Coexistence with Existing Gateways

This chapter is divided into two sections. The first section describes the architectural differences between MPM and OSDI with emphasis on administration and configuration considerations when migrating or upgrading to the Oracle9i version of the gateway. The next section describes how you migrate from a pre-Oracle9i MPM gateway to the new Oracle9i OSDI gateway.

- [OSDI Differences](#) on page 11-2
- [Migration and Upgrade](#) on page 11-7

OSDI Differences

OSDI is new Oracle software specific to IBM OS/390. It provides an OS/390 execution environment for certain Oracle products. Prior to OSDI, TG4DB2 and Oracle Net were supported by separate subsystems called MPM and TNS, respectively. OSDI completely replaces both MPM and TNS with new OS/390-specific software. The supported Oracle products are the same, however, and their generic product behavior (as viewed by application programs) is unchanged.

With OSDI, OS/390 subsystems are no longer associated with any single Oracle product or product instance. An OSDI subsystem can support multiple Oracle database instances, multiple gateway instances, and multiple Oracle Net services.

Summary of Changes

The following sections discuss the significant changes between an MPM and an OSDI configuration.

OSDI Subsystems

With OSDI, OS/390 subsystems are no longer associated with any single Oracle product or product instance. An OSDI subsystem can support multiple Oracle database instances, multiple Oracle Net services, and multiple gateway services. An OSDI subsystem has no associated address space. Refer to ["Configuring and Initializing an OSDI Subsystem"](#) on page 11-3 for a detailed description of the differences when configuring and initializing an OSDI subsystem.

Gateway Service

An OSDI gateway instance runs under the control of the OSDI subsystem. See ["Configuring a Gateway Service"](#) on page 11-4 for a detailed description of the differences as well as special considerations when running an Oracle gateway under OSDI.

Oracle Net or Network Service

The Network Service replaces the TNS facility and also runs under the control of the OSDI subsystem. The OSDI network service simplifies the networking implementation of the Oracle database service on OS/390 and is implemented to be very similar to Oracle Net on other platforms. Each gateway no longer has to maintain its own listener as a network endpoint. See the appropriate section in this

chapter for a detailed description of the differences as well as special considerations when running Oracle Net under OSDI:

- [Configuring Network Service](#) on page 11-6
- [Operating Network Service](#) on page 11-6
- [Computer Associates or Interlink SNS/TCPaccess Support](#) on page 11-7
- [IXCF Support](#) on page 11-7
- [Using Network Service](#) on page 11-7

The SQL*Net V1 style cross memory connect string (for example, W:sid) has been desupported with this release of Oracle9i for OS/390. Refer to the "Migration Considerations" chapter of the *Oracle9i Enterprise Edition User's Guide for OS/390* for further details.

Commands and Messages

With OSDI, both MPM and TNS have been superseded. All MPM-specific and TNS-specific commands and messages are obsolete.

Error Diagnosis and Reporting

The following items discuss the differences in the way the gateway generates trace and log output.

Trace Files Under OSDI, trace files can be written to spool files or disk data sets, as under MPM. This is specified using the TRACE_DSNAME gateway region parameter, see "[TRACE_DSNAME | TDSN](#)" on page 5-20 for a detailed description of the TRACE_DSNAME parameter.

Alert Logs Alert logs are written to a spool or disk file as specified by the SYSPRINT DD statement. If the SYSPRINT DD statement is omitted, then OSDI dynamically allocates a spool file for the alert log during service startup and writes a message to the system log identifying the system-generated DDname for the allocation. If you specify a disk data set for SYSPRINT, and if an error occurs while it is being written (including an out-of-space condition), then OSDI closes SYSPRINT, dynamically allocates a spool file, and begins writing to it.

Configuring and Initializing an OSDI Subsystem

The collection of the gateway instance, database instances, and network services managed by an OSDI subsystem is called a service group. OSDI subsystems are

dynamic OS/390 subsystems that do not have to be initialized at system IPL and can be initialized at any time. Once initialized, the OSDI subsystem remains in the OS/390 system until the next IPL. As a dynamic subsystem, an OSDI subsystem can be activated and deactivated at will.

In contrast to MPM and TNS, OSDI-managed services cannot be executed as OS/390 batch jobs or as independent started tasks, or STCs. Refer to ["Post-Installation Steps"](#) on page 4-14 for details on configuring and initializing an OSDI subsystem.

When migrating from MPM, avoid using the same subsystem name as any MPM or TNS subsystem that you have been running, even if you will not be using those subsystems after the OSDI installation. Unlike MPM and TNS, local OSDI clients do not normally need to be given, or need to specify the OSDI subsystem name when connecting to a server. Reusing an MPM or TNS subsystem name therefore provides no particular benefit.

Configuring a Gateway Service

To run an Oracle gateway instance under OSDI, you must define the instance as a service using the OSDI DEFINE SERVICE command. You do this when converting an existing MPM-based instance to OSDI. In addition to defining the service, a few other items must be set up before the service can be started: a JCL procedure, several parameter files, and possibly security resource definitions. After these are in place, you can start the service, which creates one or more address spaces based on controls that you have specified.

SID

Because OSDI gateways are not implemented as individual subsystems, OSDI gateways are no longer identified by subsystem names; instead, they are identified by SIDs. The SID is the OSDI Service Identifier that is used to identify the gateway. Under OSDI, the SID is defined during gateway service definition. A gateway service must be defined with a service name and optionally a SID. The service name is the default for the SID if the SID is not specified.

Gateway Instance JCL

A JCL procedure name is specified during gateway service definition. The JCL procedure must then be created before the gateway service is started. Note that JCL and load library changes should not be made while a gateway service is active. The PARM parameter of the EXEC JCL statement is not used by the OSDI gateway instance program.

The new ORA\$FPS DD statement specifies an input file containing OSDI-specific file management parameters. Refer to "[File Processing Considerations](#)" on page 11-5 for further details.

Oracle Net Access

OSDI simplifies the networking implementation on OS/390 and makes it behave in a manner similar to Oracle Net on other platforms. The Oracle Net master task MPMTNS is no longer needed in any of the OSDI gateway address spaces in order to access the OSDI gateway. For further details on configuring Oracle Net under OSDI, Refer to [Chapter 6, "Oracle Net"](#). Further descriptions of Oracle Net differences under OSDI can be found in the Oracle Net or Network Service section of this Chapter.

File Processing Considerations

OSDI processing of OS/390 data sets that are used by the gateway differs significantly from that in MPM. The major visible changes include the following:

Allocation (creation) specifications and other file processing controls are supplied to the gateway in an input parameter file, the ORA\$FPS DD that is described on page 5-32. Keyword parameters are used to specify, by type of file, things such as SMS classes, allocation unit and volume, and so forth. Each of the major types of files that is used by the gateway can be separately managed: gateway trace data sets and network trace data sets.

Operating a Gateway Service

OSDI operating commands are used to start, stop, and display OSDI services. These commands can be issued via the OS/390 command interface, either automatically (for example, COMMANDxx member of SYS1.PARMLIB) or manually via a console interface such as SDSF. All MPM commands and messages are obsolete and superseded by OSDI commands and messages. There is no OSDI equivalent to the MPMCMD command for TSO. A console interface such as SDSF must be used to issue OSDI commands from TSO. OSDI commands may also be issued through the OSDI program interface by a management component such as Oracle Enterprise manager agent. Operating commands are also permitted in the service group configuration file.

The OSDI START command must be used to start the gateway service. The OSDI START command is used to cause a gateway service to begin execution using the JCL procedure specified in the service definition. The OSDI START command (not the OS/390 command) must be used to start services. Unlike MPM, OSDI-managed

services cannot be executed as OS/390 batch jobs or as independent started tasks or STCs.

Oracle Net or Network Service

The network service replaces the TNS facility and is started with an OSDI START command. The OSDI network service simplifies the networking implementation of the Oracle gateway service on OS/390 and is implemented to be very similar to Oracle Net on other platforms.

Each gateway instance no longer has to maintain its own listener as a network endpoint. As a result, each gateway instance does not represent a distinct network address (for example, TCP/IP hostname and port) that a client had to know in order to connect to that gateway. Clients connect to target OS/390 gateway in the same manner as they connect to Oracle servers on other platforms by simply specifying the SID parameter that is part of the Oracle network address string.

Configuring Network Service

The Network Service is configured as an OSDI service using the OSDI DEFINE SERVICE command. Like other OSDI services, the Network Service is defined with a service name and SID (defaults to service name). Unlike TNS, the Network Service SID is used to identify the Network Service to inbound clients. The Network Service SID must be four characters or less. The Oracle Net JCL procedure name must have an associated OS/390 userid in order to use TCP/IP, which is controlled by OS/390 UNIX System Services (USS). The associated OS/390 userid must have an OMVS RACF segment (or equivalent, if a product other than RACF is used) if the installation is not using a default OMVS segment.

Operating Network Service

The OSDI START command must be used to start the network service. The OSDI START command is used to cause a network service to begin execution in its OS/390 address spaces using the JCL procedure specified in the service definition. The OSDI START command (not the OS/390 command) must be used to start services. Unlike TNS, OSDI-managed services cannot be executed as OS/390 batch jobs or as independent started tasks or STCs. All TNS commands and messages are obsolete and superseded by OSDI commands and messages. Refer to [Chapter 6, "Oracle Net"](#) for details on operating Oracle Net Network Services.

Computer Associates or Interlink SNS/TCPaccess Support

SNS/TCPaccess is no longer supported as a separate Oracle Net driver, support is provided via the HPNS interface of the IBM TCP/IP driver.

IXCF Support

IXCF is no longer supported as a separate Oracle Net for OS/390 protocol. IXCF can be supported via IBM TCP/IP; in which case, Oracle Net IBM TCP/IP protocol can be used.

Using Network Service

Unlike TNS, the OSDI Network service will listen for multiple OSDI Oracle gateways on the same port number, but cannot listen on multiple ports. The `tnsnames.ora` file entries that are being used by remote clients may need to be changed in order to reflect a new port number. Also for OSDI, the gateway SID must be specified in the connect string for all inbound connections. Please see [Chapter 6, "Oracle Net"](#) for a description of configuring and administering Oracle Net with OSDI. Also refer to the migration chapter of the *Oracle9i Enterprise Edition User's Guide for OS/390* for detailed discussion of migration considerations for using Oracle Net with OSDI.

Migration and Upgrade

This section describes how to migrate from MPM based gateways to the OSDI gateway. Moving to Oracle with OSDI from MPM-based Oracle and TNS-based Net is straightforward for most installations. This section of the chapter outlines the considerations in such a move, with references to detailed topic coverage in other chapters of this manual and in *Oracle9i* for OS/390 documentation.

Release Incompatibilities

This section lists the gateway behavior differences between releases.

Local Database Links

Local cross-memory database link access between an OSDI server and an MPM-based server or gateway is not supported. Access between local or remote OSDI and MPM servers or gateways must be performed via MPM-based SQL*Net and OSDI-based Oracle Net.

Migration and Upgrade Steps

Here are the steps required to upgrade or migrate your existing MPM-based gateway to the new Oracle9i OSDI-based Transparent Gateway for DB2. When possible, you should follow the links back to the detailed description of each step.

Step 1: Create and Configure an OSDI Subsystem

Before any other upgrade steps will work, the subsystem must be defined to OS/390. This step is described in detail in ["Post-Installation Steps"](#) on page 4-14, and ["Step 10: Gateway Service Definition"](#) on page 5-29.

Step 2: Create an OSDI Gateway Service

Create a JCL procedure for starting the service and place it in a system procedure library PDS. (The procedure library must be one that can be accessed without a JCLLIB statement.) Make sure that an appropriate OS/390 userid will be associated with address spaces that are started by using the procedure. Create the parameter files for the gateway program and save them as members of a PDS or as DSORG=PS data sets. Requirements for the JCL and for the region parameters are described in ["Step 11: Setup Gateway JCL"](#) on page 5-31 and ["Step 9: Edit the PARMLIB Members"](#) on page 5-18.

Step 3: Create and Configure OSDI Net Service

Create and configure the OSDI Net Service as described in [Chapter 6, "Oracle Net"](#). Pay particular attention to the TNSNAMES entries, be sure entries are updated to include SID parameters.

Step 4: Establish Security

Perform installation required security steps for the OSDI subsystem and the gateway to DB2 connections. The OSDI security steps are covered in ["Step 2: Associate Userids with Services"](#) on page 4-20. The gateway to DB2 security steps will depend upon your current connection level security situation.

If your current MPM based gateway used the provided security exits, G4AUTH and DB2AUTH as provided (i.e. without modification), simply follow the steps documented in ["Gateway Security"](#) on page 7-4, and ["Step 3: Run the sidJC01 and sidJC02 Configuration Jobs"](#) on page 5-11. If your installation has modified DB2AUTH, this will continue to work. There is no need to re-modify or reassemble. However, if your installation has modified the provided G4AUTH, the modifications will have to be added to the new G4AUTH and a reassembly performed.

Step 5: Ensure User Exits are Available to DB2

In addition to the security exits discussed above, you must ensure that the date user exits are also available to the new gateway if they've been implemented in your current MPM based gateway. This procedure is covered in ["Step 4: Make Authorization and Local Date Exits Available to DB2"](#) on page 5-11.

Step 6: Prepare the DB2 Environment

The DB2 environment must now be upgraded so that the Oracle packages, plan, views, and tables are at the level expected by the gateway. The necessary steps are documented in detail at:

[Step 5: Run the Scripts to Create Required Tables and Views in DB2](#) on page 5-12

[Step 6: Bind the DB2 Package](#) on page 5-12

[Step 7: Bind the DB2 Plan](#) on page 5-15

[Step 8: Grant EXECUTE on DB2 Plan](#) on page 5-18

Step 7: Start the OSDI Services

Start the OSDI services defined in STEPs 2 & 3, as described in ["Operation of the Gateway Subsystem with OSDI"](#) on page 7-2.

Step 8: Test the Gateway

The last step is to create the database link as described in ["Database Link Behavior"](#) on page 8-1, and test the connection between Oracle, the gateway, and DB2.

Configuring Multiple OSDI Gateway Services

You can create multiple gateway services simply by defining a new service with a different service name and a different SID. The gateways will share the same subsystem, but all parameters, including the DB2 region they connect to, can be different. The steps required are a subset of the above steps.

The first step is the same as Step 2 above, paying particular attention that the service name and the SID are unique.

If your new gateway service is connecting to the same DB2 region as your existing gateway, skip to Step 7 above. If not, skip to Step 4 and execute each step as before.

MPM/TNS and OSDI Coexistence

You can continue to run MPM-based Oracle instances and TNS network services during your transition to OSDI. However, you must be aware of the limitation with database links (distributed access) between Oracle and/or gateway instances on the same OS/390 system. Cross-memory database links (in either direction) between MPM and OSDI instances are not supported. If you have a requirement for such distributed access, then you will need to use TCP/IP instead of cross-memory access. Doing this requires running both OSDI Oracle Net and TNS with distinct TCP endpoint definitions.

TNS and OSDI NET Communication

An example of this type of MPM to OSDI connection is as follows:

The TNSNAMES entry defined to the OSDI Oracle instance to access the MPM based gateway is:

```
G4XXMPM =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=TCP) (HOST=MVS.TCP.IP.ADDR) (PORT=3998)
    (SSN=NET))
    (HS=))
```

Where PORT would be the port number that MPM TNS is listening on for the MPM gateway, and SSN is the SID of the OSDI NET.

The TNSNAMES entry defined to the MPM Oracle instance to access the OSDI based gateway is:

```
G4XXOSDI =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=TCP) (HOST=MVS.TCP.IP.ADDR)
    (PORT=4998) (SSN=TNS)
    (HS=)
    (CONNECT_DATA=(SID=G4X1))
```

Where PORT would be the port number that OSDI NET is listening on for all defined OSDI services in its service group, SID is the SID of the OSDI gateway, and SSN is the SSN of the MPM TNS.

The DBLINK defined to the MPM based Oracle instance to access the OSDI based gateway would be:

```
Create database link g4osdi connect to scott identified by
tiger using 'G4XXOSDI';
```

The DBLINK defined to the OSDI based Oracle instance to access the MPM based gateway would be:

```
Create database link g4mpm connect to scott identified by  
tiger using 'G4XXMPM';
```

Data Set Names and Space Allocations

This appendix provides information about the NEWDSRPT CLIST and Oracle default data sets. The following topics are included:

[Using the NEWDSRPT CLIST](#) on page A-2

[Oracle Default Data Sets](#) on page A-2

Using the NEWDSRPT CLIST

During installation, the NEWDSRPT CLIST is loaded into your Oracle ISPCLIB library. This CLIST produces a report, as a selected PDS member, listing the default library sizes for the data sets corresponding to the products you selected during installation.

When you run this CLIST, it prompts for:

- High-level and second-level qualifiers for your Oracle ISPTLIB library
- High-level and second-level qualifiers for your selected output library (the low-level qualifier is DSPRT)
- PDS member name for the report

The CLIST writes the report to your selected member in the high-level *oran.orav.DSPRT* library. The output PDS must already exist for the CLIST to execute successfully. The output PDS can be fixed block (FB) or fixed-block-access (FBA) and have a logical record length of 80 or 133.

TSO message ***File DSPRT not freed, is not allocated*** is an informational message and does not indicate an error condition.

Oracle Default Data Sets

If you choose to install all products on this tape, member DFLTLIST in the Oracle ISPSLIB library lists the target data sets that are created during the Oracle Transparent Gateway for DB2 for OS/390 installation. It also provides the default space allocations, in tracks, for each data set.

Member SHIPLIST in the Oracle ISPSLIB library contains a list of all the Oracle product data sets that are shipped on the distribution media with their respective space allocations. If you do not install all the products on this tape, you can use this member to determine how your data set space allocations differ from the default list provided in member DFLTLIST.

OSDI Subsystem Command Reference

OSDI provides a set of system commands for defining and controlling instances of Oracle products. This appendix is the primary reference for all OSDI commands.

The topics in this appendix include:

- [Command Types and Processing](#) on page B-2
- [System Symbols in Commands](#) on page B-2
- [Definition Commands](#) on page B-3
- [Structures](#) on page B-3
- [Service Group Definition Commands](#) on page B-3
- [Service Definition Commands](#) on page B-5
- [Operating Commands](#) on page B-11
- [Available Commands](#) on page B-11
- [Commands](#) on page B-11
- [OSDI Command Keyword Abbreviations](#) on page B-15

Command Types and Processing

Commands are broadly divided into two groups: definition and operating.

1. Definition commands are used to create and manipulate data structures that describe service groups and services. These commands commonly appear in the subsystem configuration file.
2. Operating commands are used to manage execution of services.

Three mechanisms exist for issuing OSDI commands:

1. Subsystem configuration file - processed during subsystem initialization.
2. OS/390 system operator command interfaces-system consoles, TSO SDSF, SYS1.PARMLIB (COMMNDxx), Netview, and others.
3. OSDI program interface - used internally by Oracle products.

When an OSDI command is issued using an OS/390 system operator command interface, the target subsystem is specified by using the command prefix associated with the subsystem. When the program interface is used to issue an OSDI command, the target is identified by its subsystem name rather than a command prefix. Commands in the configuration file always apply to the subsystem (service group) being initialized, and they must omit the prefix.

Commands generally result in synchronous response messages ranging from simple acknowledgment to multiline displays. Responses to commands that are issued via system operator facilities normally are directed to the issuing console. Various operating commands can result in subsequent, asynchronous messages. These messages are not necessarily directed to the console or session that issued the original command.

System Symbols in Commands

In order to meet the requirement that the particulars of a service (that runs on multiple systems in a sysplex) can be tailored by system, OS/390 system symbols (alphanumeric names prefixed with the ampersand character, "&") can be used in the specification of certain OSDI command parameters. These command parameters resolve to system-specific or IPL-specific values set by OS/390 or by the installation. If a command parameter can include system symbols, this capability is noted in the parameter description.

Definition Commands

Definition commands are used to create, modify, and display the data structures of the service group. An initial set of commands in the configuration file directs the building of these structures during subsystem initialization. Subsequent definition commands can be used to add new service definitions, modify existing definitions, and so forth. The overall data structure persists for the life of the IPL.

The definition commands operate only on data structures; they do not directly affect the operation of services.

Three definition commands are supported:

- **DEFINE** - Create a logical structure.
- **ALTER** - Modify the definition of an existing logical structure.
- **SHOW** - Display contents of an existing logical structure.

Note: **SHOW** deals with definition data only and is distinct from the operating command, **DISPLAY**. Refer to "**SHOW**" on page B-5 and "**DISPLAY**" on page B-12.

Structures

The definition commands operate on the following structures:

- **SERVICEGROUP** - The primary structure of the subsystem.
- **SERVICE** - A structure representing an instance of an Oracle product.

The various parts of these structures are generally referred to as attributes. Definition commands use keywords to identify attributes being set or modified.

Service Group Definition Commands

The Service Group Definition commands are described in the following paragraphs.

DEFINE

DEFINE SERVICEGROUP must be the first command issued to a newly-initialized subsystem; normally it appears in the configuration file just after the bootstrap (INIT) record. **DEFINE SERVICEGROUP** must be processed successfully in order for any other OSDI commands or functions to be usable.

The command structure for defining a service group is shown in the following example:

```
DEFINE SERVICEGROUP
  [ SSID( string ) ]
  [ DESCRIPTION( string ) ]
  [ MODE( LOCAL | SYSPLEX | *SYS ) ]
  [ SYSTEMS( sysname [ sysname... ] | *ALL ) ]
```

Define Parameters

SSID	Specifies the 1-character to 4-character OS/390 subsystem name associated with the service group. If coded, then it must match the subsystem identifier specified in the IEFSSNxx parmlib member or the SETSSI operator command. This parameter defaults to the correct value. It is therefore coded only to confirm that the correct configuration file is in use. <i>String</i> cannot contain system symbols.
DESCRIPTION	Specifies an arbitrary text string of up to 64 characters that appears in certain displays associated with the service group. This can be used to supply installation-specific identification for the service group. The default value is 'Service Group <i>ssn</i> ' where <i>ssn</i> is the subsystem name. <i>String</i> can contain system symbols but should not contain non-printable characters or control characters.
MODE	This parameter is not yet supported.
SYSTEMS	This parameter is not yet supported.

ALTER

ALTER SERVICEGROUP is used to modify attributes of a service group. It can be included in the configuration file and it can be issued after initialization. Not all attributes can be altered. The subsystem ID, for example, is constant for the life of the IPL.

The command structure for altering a service group is shown in the following example:

```
ALTER SERVICEGROUP
  [ DESCRIPTION( string ) ]
```

```
[ MODE( LOCAL | SYSPLEX ) ]
[ SYSTEMS( sysname [ sysname... ] | *ALL ) ]
```

Alter Parameters

DESCRIPTION	Specifies an arbitrary text string of up to 64 characters that appears in certain displays associated with the service group. This can be used to supply installation-specific identification for the service group. <i>String</i> can contain system symbols but should not contain non-printable or control characters.
MODE	This parameter is not yet supported.
SYSTEMS	This parameter is not yet supported.

SHOW

SHOW SERVICEGROUP is used to display the current definition of the service group. It can be included in the configuration file and it can be issued after initialization. The command for displaying a service group definition is shown in the following example:

```
SHOW SERVICEGROUP
  [ LONG ]
```

Show Parameters

LONG	Specifies that the name, type, and description of each service in the service group be included in the display.
------	---

Service Definition Commands

The Service Definition commands are described in the following paragraphs.

DEFINE

DEFINE SERVICE is used to create a structure for executing an installed Oracle product. It can be included in the configuration file and it can be issued after initialization. Once a service is defined, it can be started, stopped, etc. using operating commands.

The command structure for defining a service is shown in the following example:

```
DEFINE SERVICE
    name
    TYPE( string )
    PROC( string )
    [ DESCRIPTION( string ) ]
    [ PARM( string ) ]
    [ MAXAS( number ) ]
    [ SID( string ) ]
    [ JOBNAME( string ) ]
    [ JOBACCT( string ) ]
    [ MODE( SYSPLEX | LOCAL ) ]
    [ SYSTEMS( sysname [ sysname... ] | *SVG ) ]
```

Define Parameters

name	<p>Specifies the name of the service. The name is used in other commands that operate on the service or its definition and by program functions that interact with the service. It must be 1 character to 8 characters long, must consist of upper case alphabetic, numeric, and/or national characters, and must begin with an alphabetic character. It must be unique within the service group. <i>Name</i> cannot contain system symbols.</p> <p>Note: Unless you specify <i>JOBNAME</i>, <i>name</i> is used as the job identifier when the service is started. In this case, <i>name</i> must not be the same as any subsystem name in your system.</p>
TYPE	<p>Specifies the Oracle product being configured. <i>String</i> is a 1-character to 4-character alphabetic and/or numeric identifier that designates an installed Oracle for OS/390 product managed under this architecture. Current supported values are ORA (Oracle database), GTW (Oracle Gateway), and NET (Oracle Net). <i>String</i> cannot contain system symbols.</p>
PROC	<p>Specifies the member name in a system JCL procedure library used to start an address space for the service. Usually this is a procedure that is created during installation of the associated Oracle for OS/390 product. <i>String</i> cannot contain system symbols.</p>

DESCRIPTION	Specifies an arbitrary text string of up to 64 characters that appears in certain displays associated with the service. This can be used to supply installation-specific identification for the service. The default value is 'Service <i>svc</i> Type <i>type</i> ', where <i>svc</i> is the service name and <i>type</i> is the type. <i>String</i> can contain system symbols but should not contain non-printable or control characters.
PARM	The PARM for a gateway service specifies the name of an OS/390 data set containing service initialization parameters. These are OS/390-specific parameters and are described in the section Step 9: Edit the PARMLIB Members on page 5-18.
MAXAS	Specifies the maximum number of address spaces that can be started for the service. This is meaningful only for a database (TYPE(ORA)) service. If specified for any other type, then <i>number</i> must be 1. For a database service, <i>number</i> must be between 1 and 255 inclusive. The default for this parameter is 1. The value cannot include system symbols.
SID	<p>Specifies a unique identifier for the service that is used in client- or application-supplied service addressing. <i>String</i> is a 1-character to 8-character identifier that conforms to the same rules as those for service names. The value supplied must be unique within the OS/390 image: it cannot duplicate the SID of any other service in this or any other service group. This parameter defaults to the service name. If you accept the default, it means that the service name must be unique within the OS/390 image.</p> <p>Note: If you migrate an MPM-based gateway to OSDI and are supporting pre-OSDI local client applications, you probably want SID to match the subsystem name you were using with MPM.</p>

JOBNAME	Specifies an OS/390 jobname to use when starting address spaces for the service. <i>String</i> is either a 1-character to 8-character identifier that conforms to OS/390 jobname requirements, or it is a 1-character to 5-character identifier followed by an asterisk. The asterisk form can be used with multiple address space services to provide unique jobnames; it is replaced with a 3-digit address space counter (001, 002, and so on) as each address space is started. The jobname should conform to any rules or requirements specific to your installation. This parameter has no default. If you omit it, service address spaces are started without a jobname but with the service name as the address space identifier. System symbols cannot be used in this parameter.
JOBACCT	Specifies an installation-specific string containing job accounting fields. <i>String</i> can be from 1 character to 64 characters. If it contains characters other than alphabetic, numeric, and national characters, then it must be enclosed in single apostrophes. Use this parameter if your installation requires job accounting data to be included with started tasks (STCs). If this parameter is omitted or specified as a null string (a pair of adjacent apostrophes), then no job accounting data is supplied when service address spaces are started. <i>String</i> can include system symbols.
MODE	This parameter is not yet supported.
SYSTEMS	This parameter is not yet supported.

Refer to ["Examples"](#) on page 4-19.

ALTER

ALTER SERVICE is used to modify attributes of a defined service. It can be included in the configuration file and it can be issued after initialization. The name, type, maximum address spaces, and SID of a service cannot be altered.

OSDI does not prohibit altering the definition of a running service. This enables some useful capabilities but it may also be harmful if misused. For example, changing the JCL procedure of a running multiple address space service would have unpredictable consequences when additional auxiliary address spaces are started.

The command structure for altering a service is shown in the following example:

```
ALTER SERVICE
    name
    [ DESCRIPTION( string ) ]
    [ PROC( string ) ]
    [ PARM( string ) ]
    [ MAXAS( number ) ]
    [ JOBNAME( string ) ]
    [ JOBACCT( string ) ]
    [ MODE( SYSPLEX | LOCAL ) ]
    [ SYSTEMS( sysname [ sysname... ] | *SVG ) ]
```

Alter Parameters

name	Specifies the name of the service to be altered. It must be the name of an existing service in the service group. <i>Name</i> cannot contain system symbols. Note: The name of a service cannot be altered.
DESCRIPTION	Specifies an arbitrary text string of up to 64 characters that appears in certain displays associated with the service. This can be used to supply installation-specific identification for the service. <i>String</i> can contain system symbols but should not contain non-printable or control characters.
PROC	Specifies the member name in a system procedure library used to start an address space for the service. Usually this is a procedure that is created during installation of the associated Oracle for OS/390 product. <i>String</i> cannot contain system symbols. Note: Altering PROC while a multiple address space or multiple system service is running can have unpredictable effects.

PARM	<p>Specifies a parameter string passed to the service when an address space is started. Requirements for this string depend on the service type and are discussed in the associated Oracle for OS/390 product documentation. The value specified can contain system symbols.</p> <p>Note: Altering PARM while a multiple address space or multiple system service is running can have unpredictable effects.</p>
MAXAS	<p>Specifies the maximum number of address spaces that can be started for the service. This is meaningful only for a database (TYPE(ORA)) service. MAXAS is accepted on an ALTER SERVICE command only when the service is not active. If the service is active, starting, or stopping, an ALTER SERVICE command with MAXAS specified will be rejected.</p>
JOBNAME	<p>Specifies a jobname to use when starting service address spaces, as discussed under DEFINE SERVICE. You can nullify (remove) a service jobname specification by coding JOBNAME(""). The value specified cannot contain system symbols.</p>
JOBACCT	<p>Specifies job accounting data to be included when service address spaces are started, as discussed under DEFINE SERVICE. You can nullify (remove) job accounting data by coding JOBACCT(""). The value specified can contain system symbols.</p>
MODE	<p>This parameter is not yet supported.</p>
SYSTEMS	<p>This parameter is not yet supported.</p>

SHOW

The SHOW SERVICE command is used to display the current definition of a service.

The command for displaying a service definition is shown in the following example:

```
SHOW SERVICE
      name
```

Show Parameters

name Specifies the name of the service whose definition is to be displayed. It must be the name of an existing service in the service group. *Name* cannot contain system symbols.

Operating Commands

Operating commands manage the execution of services. They are normally issued via the OS/390 command interface, either automatically (for example, COMMNDxx member of SYS1.PARMLIB) or by a real operator. They might also be issued through the OSDI program interface by a management component such as Oracle Enterprise Manager Agent. Operating commands are also permitted in the service group configuration file.

Available Commands

Five operating commands are provided:

- **START** - Start execution of a service.
- **DISPLAY** - Display operating status of a service.
- **DRAIN** - Place a service in quiescent state.
- **RESUME** - Restore a service to normal operation.
- **STOP** - Stop execution of a service (see note below).
- **KILL** - To terminate a dedicated task.

All of the operating commands take a service name as the first positional parameter. This service name must be the name of a defined service.

Commands

The commands are described in the following paragraphs.

START

The **START** command initiates execution of an address space for a specified service. For a database service, this can be the first address space (service not previously

started) or an auxiliary address space (service previously started and initialized successfully but not yet at its maximum address spaces). For other types of services the service must not already be running.

The command structure for starting a service is shown in the following example:

```
START
    name
    [ PARM( string ) ]
```

START Parameters

name *Name* specifies the name of the service to start. It must be a defined service whose current state is **inactive** or, if **active**, must not already have its maximum address spaces running.

PARM Specifies a parameter string passed to the service when an address space is started. This overrides any PARM value established by DEFINE or ALTER SERVICE. Requirements for this string depend on the service type and are discussed in [Chapter 6, "Oracle Net"](#). *String* can contain system symbols.

Note: A PARM override must not be used when starting auxiliary address spaces for a database service.

DISPLAY

The DISPLAY command displays execution status information for services. OSDI displays the current operating state of the service. If the service state is "active" or "drained", then the command is also posted to the running service for further processing at the service's discretion.

The command structure for displaying a service is shown in the following example:

```
DISPLAY
    name
    [ LONG ]
```

DISPLAY Parameters

name Specifies the name of the service to be displayed.

LONG Specifies that a more detailed display is desired. This provides information about each active address space of the service.

DRAIN

The DRAIN command places a running service in a quiescent state in which it no longer accepts new connection (bind) requests. Existing connections or sessions are not affected. The command is also posted to the running service for further discretionary processing. This command has no effect when the service is not running.

The command structure for draining a service is shown in the following example:

```
DRAIN
  name
```

DRAIN Parameters

name Specifies the name of the service to be made quiescent. This must be the name of a running service whose current state is **active**.

RESUME

The RESUME command reverses the effect of a drain, allowing a service to begin accepting new connection requests. The command is also posted to the running service for further discretionary processing. This command has no effect when the service is not running.

The command structure for resuming a service is shown in the following example:

```
RESUME
  name
```

RESUME Parameters

name Specifies the name of the service to be resumed. This must be the name of a running service whose current state is **drained**.

STOP

The STOP command requests termination of a running service. The normal mode of STOP changes the service state to **stopping** and posts the stop request to the service; it is up to the service to comply, presumably after allowing current requests to complete and performing required cleanup. A **FORCE** option causes the service address space(s) to be terminated involuntarily. The **FORCE** form of STOP requires that a normal STOP be issued first. This command has no effect when the service is not running.

The command structure for stopping a service is shown in the following example:

```
STOP
    name
    [ FORCE ]
```

STOP Parameters

name	Specifies the name of the service to be stopped. This must be the name of a running service whose current state is active , drained , or (if the FORCE option is specified) stopping .
FORCE	Specifies that an involuntary stop is requested.

KILL

The KILL command terminates a dedicated task. It's a useful command to terminate an idle gateway session into DB2. DB2 has a strict locking model and very often leads to a deadlock situation. In order to release the DB2 locks due to long idle dblink session, you could use the OSDI DISPLAY command to identify the desired idle session; and use this KILL command to terminate the dedicated task (i.e., gateway session into DB2).

The command structure for killing a service is shown in the following example:

```
KILL SESSION(kkkkkkkk)
```

Where *kkkkkkkk* is the 8-digit (hex) session key.

Example:

```
F service_name,D SESS JOB(*)
MIR0503I Sess=00050010, Job=RSTETAK3, TCB=007B0E78, AS=001, User=SCOTT
MIR0500I Command processed
```

```
F service_name,KILL SESSION(00050010)
MIR0500I Command processed
```

OSDI Command Keyword Abbreviations

The abbreviated or alternate forms that can be used for OSDI command verbs and parameter keywords are as follows:

ALTER	ALT
DEFINE	DEF
DESCRIPTION	DESC
DISPLAY	DIS, D
DRAIN	DR
FORCE	(none)
JOBACCT	ACCT
JOBNAME	JOB
LONG	L
MAXAS	MXA
MODE	MD
PARM	P
PROCEDURE	PROC
RESUME	RES
SERVICE	SRV, SVC
SERVICEGROUP	SVG, SG
SHOW	SH
SID	IDENTIFIER, ID
SSID	(none)
START	ST, S
STOP	P
SYSTEMS	SYS
TYPE	TY

National Language Support

National Language Support (NLS) is a technology enabling Oracle applications to interact with users in their native language, using their conventions for displaying data.

The following topics are included:

[Overview](#) on page C-2

[Default Character Set Changes](#) on page C-2

[Oracle9i Enterprise Edition for OS/390](#) on page C-2

[Gateway Configuration](#) on page C-3

[Supported Character Sets](#) on page C-4

[Oracle Database Server and Client Configuration](#) on page C-8

[Message Availability](#) on page C-8

Overview

The Oracle NLS architecture is data-driven, enabling you to add support for specific languages and character encoding schemes without requiring any changes in source code.

A number of different settings in the gateway, DB2, Oracle database server, and client affect NLS processing. Character settings of these components must be compatible so that translations occur correctly. Each character in one encoding scheme must have a matching character in the other encoding schemes.

Default Character Set Changes

At installation, the gateway character set defaults to WE8EBCDIC37C to ensure compatibility with DB2. DB2 does not support WE8EBCDIC1047.

The language-related parameters should always match the DB2 coded character set identifier (CCSID). If changes are made to the default character set name, then it is important that the new parameter matches the DB2 CCSID. For details, refer to [Table C-1, "National Language Support for the Gateway"](#) on page C-4.

If you are using Oracle9i Enterprise Edition for OS/390 as the integrating server, then refer to the next section, "[Oracle9i Enterprise Edition for OS/390](#)", for further details prior to installing the gateway and Oracle9i Enterprise Edition for OS/390.

Oracle9i Enterprise Edition for OS/390

Beginning with Oracle Enterprise Edition for OS/390 release 7.3.2, the Oracle database server is created with the default character set WE8EBCDIC1047 to facilitate compatibility with the IBM OS/390 compiler character set.

When Oracle Enterprise Edition for OS/390 release 7.3.2 or later is used as the integrating server in your Oracle Transparent Gateway for DB2 configuration, it is important (when creating the Oracle9i Enterprise Edition for OS/390 database) to consider using the same character set for Oracle9i Enterprise Edition on OS/390 that has been defined during the gateway installation. In most cases, the gateway character set reflects WE8EBCDIC37C. It is important to confirm this setting because the default setting can be changed.

If the character sets vary between Oracle9i Enterprise Edition for OS/390 and the gateway, then performance can be impacted because all data requires translation. In addition, the NLS_LANG specification of an OS/390 client process (such as SQL*Plus or client application programs) should be the same for the Oracle9i

Enterprise Edition for OS/390 database to avoid translation between the client process and Oracle9i Enterprise Edition for OS/390.

Gateway Configuration

Once the gateway is installed, parameters must be set correctly in the data sets specified in the ORA\$ENV DD statement in the startup JCL for the gateways.

NLS Parameters in *sid*ENV

NLS_LANG must be specified to define the character set used by the gateway. The NLS_LANG parameter must be changed in the data set specified by the ORA\$ENV DD statement.

The syntax for the NLS_LANG parameter is:

```
NLS_LANG='language[_territory.character_set]'
```

where:

language is any language listed in [Table C-2, "Supported Languages and Territories"](#).

territory is optional and defaults to AMERICA. Refer to [Table C-2, "Supported Languages and Territories"](#), for a list of supported territories.

character_set defaults to WE8EBCDIC1047. It must correspond with a character set used by the DB2 system.

Since DB2 does not support the OS/390 Oracle server default character set of WE8EBCDIC1047, you must always specify a character set that is supported by DB2, or accept the default.

NLS_NCHAR must be specified for DBCS users. The syntax is:

```
NLS_NCHAR='character_set'
```

For example:

```
NLS_NCHAR='K016DBCS'
```

DB2 Configuration

The Oracle language parameters must match the DB2 CCSID of the DB2 system installed at your site. For example, if the DB2 CCSID is 280 for Italy, then the gateway character set must be set to I8EBCDIC280. For additional information about DB2 CCSID codes, contact your database administrator and refer to the IBM documents for your platform and operating system.

Supported Character Sets

The following languages and values for `character_set` are supported by Oracle Transparent Gateway for DB2:

Table C-1 National Language Support for the Gateway

Description	character_set	DB2 CCSID
Austrian/German	D8EBCDIC273	273
Traditional Chinese	ZHT16DBCS	937
Danish/Norwegian	DK8EBCDIC277	277
Finnish/Swedish	S8EBCDIC278	278
French	F8EBCDIC297	297
Greek	EL8EBCDIC875	875
Italian	I8EBCDIC280	280
Japanese	JA16DBCS	939, 5035
	JA16EBCDIC930	930, 5026
Korean	KO16DBCS	933
Simplified Chinese	ZHS16DBCS	935
Spain	WE8EBCDIC284	284
Thai	TH8TISEBCDIC	838
Western European	WE8EBCDIC37	37
	WE8EBCDIC37C	37
	WE8EBCDIC500	500
	WE8EBCDIC500C	500

Note: If you specify JA16EBCDIC930 in a language parameter for the gateway, then you must specify NLS_DATE_FORMAT in all uppercase letters.

Double-Byte Character Support

Note: You need to apply zap B2272343, if not already applied. Contact your local Oracle support team for details.

The support of mapping GRAPHIC to var(char) is enabled with 2 environment parameters specified in ORA\$LIB DD.

ORAGRAPH4=YES is required.

The followings describe additional consideration for parameter 'ORADBMBOPT'.

This version of the Oracle Transparent Gateway for DB2 allows you to access and change columns in DB2 that are designated as VARGRAPHIC or GRAPHIC. Because of the IBM definition of a (VAR)GRAPHIC column, you can insert only double-width characters into (VAR)GRAPHIC columns—in other words, the string to be inserted must consist entirely of DBCS characters. Attempts to insert a mixed-byte string into a (VAR)GRAPHIC column will result in an error. For example, attempting to insert a value represented by a mixed-byte string constant into a (VAR)GRAPHIC column will result in SQL error.

An ASCII client might attempt to use the following INSERT statement:

```
INSERT INTO mytable.graphcol@tg4db2 values('AxxB')
```

where "A" and "B" are the normal ASCII single-width characters and "xx" designates a double-width character in one of the Far Eastern ASCII-based character sets such as JA16SJIS, ZHS16CGB213280, ZHT16BIG5, or KO16KSC5601. There is no unambiguous way in which to insert such data into a (VAR)GRAPHIC column. The problem exists because no DBCS character corresponds to each of the single-byte characters.

This version of the Oracle Transparent Gateway for DB2 provides a workaround, but the workaround carries a risk for data integrity. If you desire to choose this path, then you must realize that if you attempt to INSERT a string with a single-byte character, then that string may be returned to you entirely as a DBCS string in a subsequent SELECT as a double-byte character. The rules are as follows:

1. With no options specified, the user must ensure that any strings to be INSERTed into (VAR)GRAPHIC columns are in their "wide" form on the ASCII machine. The string can not contain any single-byte characters. This means that when converting from the ASCII-based string to a DBCS string, the conversion does not result in a DBCS string with a SHIFT-OUT or SHIFT-IN character at any other position than the first character position for the SHIFT-OUT character or the last character position for the SHIFT-IN character. The same is true for values represented by constants or for values represented by bind-variables.
2. Specifying the FORCE_SB option causes all single-byte characters to be translated to their corresponding double-byte values on input—via either an INSERT or a bind-variable. The entire string is then INSERTed into the (VAR)GRAPHIC column. On output, that is to say, on a SELECT, each (VAR)GRAPHIC column will be examined for double-byte characters that correspond to single-byte characters. Each such character is replaced by its corresponding single-byte character. This is where the data integrity problem arises. Since single-byte characters have been forced to their corresponding double-byte characters on input, there is no way to know if a double-byte character actually came from the translation of an ASCII wide character or from the process of forcing single-byte characters to the corresponding double-byte characters.

As an example, use the previous INSERT statement, which is:

```
INSERT INTO mytable.graphcol@tg4db24 values ('AxxB')
```

If you use the FORCE_SB option, then the resultant DBCS value in the DB2 column may look like the following:

```
wAyywB
```

where "wA" is the DBCS correspondence (usually 0x42C1) for the single byte 'A', "wB" is the DBCS correspondence (usually 0x42C2) for the single byte 'B', and "yy" is the DBCS character corresponding to the ASCII-based 'xx'.

When SELECTing from this column, on the client you would get exactly what you INSERTed, that is, 'AxxB'—which is good. But what if the DBCS character that is represented in the DB2 column by "wa" had actually been INSERTed into the column via a valid ASCII representation for a wide A? With the FORCE_SB option, you get a single-byte A on the client. This may not be exactly what you wanted.

In the end, it is you, the customer, who must decide whether this option is valuable to you. You must decide if you can accept the possible problems that can arise.

The ORADBMBOPT environment variable controls this feature. If there is no ORADBMBOPT environment variable, then no forcing of single-byte character to double-byte character will take place when transferring data into DB2 columns, and no forcing of double-byte characters to the single-byte correspondences will take place when transferring data from DB2 columns. Placing the string "FORCE_SBCS" into ORADBMBOPT turns the feature ON.

Note that you should not use PassThrough to execute any SQL commands that contain graphic constants, unless those constants conform fully with DB2 graphic constants. DB2 graphic constants start with G' (G apostrophe) or N' (N apostrophe) and end with an apostrophe (') character. The first character after the G' or N' must be a SHIFT-OUT character and the character preceding the apostrophe (') at the end of the constant must be a SHIFT-IN character.

Between the SHIFT-OUT and SHIFT-IN characters, an even number of bytes must be present, with each byte pair making up a single DBCS character.

Supported Languages and Territories

The gateway supports the following language and territory combinations:

Table C-2 Supported Languages and Territories

Language	Default Territory
American	America
Chinese	Taiwan
Czech	Czechoslovakia
Danish	Denmark
Dutch	Netherlands
Finnish	Finland
French	France
Greek	Greece
German	Germany
Hungarian	Hungary
Italian	Italy
Japanese	Japan

Table C-2 Supported Languages and Territories

Language	Default Territory
Norwegian	Norway
Polish	Poland
Portuguese	Portugal
Slovak	Czechoslovakia
Spanish	Spain
Swedish	Sweden

Oracle Database Server and Client Configuration

There are a number of NLS parameters controlling NLS processing between the Oracle database server and client. You can set language-dependent behavior defaults for the server and 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 Server SQL Reference*. These parameters do not affect the gateway processing. However, you must ensure the character set is compatible with the character sets you specify on the gateway and DB2. In other words, each character in one encoding scheme must have a matching character in another encoding scheme.

When you create your database, the character set used to store data is specified by the CHARACTER SET parameter. Once the database is created, the database character set cannot be changed unless you re-create the database.

Normally, the default for CHARACTER SET on non-EBCDIC platforms is US7ASCII, which supports only the 26 Latin alphabetic characters. If you have specified 8-bit character sets on the gateway and DB2, then you must have a compatible 8-bit character set defined for your database. To check the character set of an existing database, enter the command:

```
SELECT USERENV('language') FROM DUAL;
```

Message Availability

Availability of the supported languages depends on which modules are installed in the Oracle product set. If you do not have message modules for a particular language set installed, then specifying that language with NLS_LANG does not show messages in the requested language.

The Oracle SMF Interface

The IBM System Management Facility (SMF) provides a facility for users to collect and record a variety of system and job-related information. SMF formats the information into a number of different records. By creating analysis and report routines, installations can use the information in SMF records to track system usage.

The gateway service uses the standard SMF interface to write user records to SMF data sets. These user records contain gateway service accounting and information allowing gateway installation sites to charge individual users for the resources they use.

DB2 also uses SMF to collect and record data. DB2 uses record types 100, 101, and 102. Refer to your DB2 documentation for more information.

The following topics are included:

[Activating SMF Records](#) on page D-2

[Events Generating SMF Records](#) on page D-3

[Interpreting an Oracle SMF Record](#) on page D-3

[ORAFMT Sample Formatting Program](#) on page D-8

Activating SMF Records

SMF recording is activated by updating the SMFPRM_{xx} member of SYS1.PARMLIB to include the SYS or SUBSYS option. These options record the gateway user record type. If the SUBSYS option is used, then the SUBSYS name must match the OSDI subsystem name specified by the OSDI startup parameter. Refer to the IBM guide on System Management Facilities for information about implementing SMF.

Specifying the Oracle Gateway Record Type

The default gateway user record type is 0 (zero). A zero for this parameter indicates that no SMF statistics record is to be written. You can override the default to any value between 128 and 255 by adding the SMF_STAT_RECNO (abbreviation is SMFSTRCN) to the OSDI gateway region parameter file. The SMF record number that is chosen must not be the same as the number that is used by any other OS/390 software.

Oracle Corporation recommends using SFM record number 199, but any available record number between 128 and 255 may be used. Due to the differences in the report formatting programs, OSDI gateways and MPM gateway should use different SMF record numbers.

If this parameter is not specified, or if zero is specified, then no SMF statistics connection or recording is done. This saves some CPU overhead and saves the overhead of the SMF write itself (which is mostly asynchronous work done by the SMF address space, and the in-line overhead is mostly just moving data into SMF buffers).

Using the OSDI SMF_STAT_RECNO Parameter

To override the default record type in the SMFPRM_{xx} member, use the following syntax:

```
SMF_STAT_RECNO | SMFSTRCN
```

SMF_STAT_RECNO can be added as an OSDI parameter to the OSDI gateway region parameter data set to override the default record number 0. In the following example, 199 is the new SMF record type:

```
SMF_STAT_RECNO(199)
```

Starting SMF Recording

SMF recording of gateway accounting information starts automatically at startup if SMF is activated and the gateway record type is specified in the SMFPRM_{xx} member.

Because the standard system default record types activated for SMF are 128 through 255, and because the recommended value for the gateway service (199) is within this range, many sites automatically begin SMF recording of gateway records when the gateway is installed.

If the SMF_STAT_RECNO is added or modified in the OSDI gateway region parameter file while the gateway is active, the service must be stopped and restarted for the parameter to take effect.

Stopping SMF Recording

The OSDI SMF_STAT_RECNO parameter can be used to stop SMF recording for Oracle. To stop SMF recording for Oracle regardless of what your system tables specify, use:

```
SMF_STAT_RECNO (0)
```

or take the default of 0 (zero). The service must be stopped and restarted for this parameter to take effect.

Events Generating SMF Records

After SMF recording is turned on, an SMF record is written each time a user logs off (normal termination or SMFINV set to SMFNORM), provided SMF is activated when the user logs on.

SMF records are also written on an abend or cancellation of a job if SMFINV=SMFABORT.

If the OS/390 system crashes, then SMF records are not written and the information is lost.

Interpreting an Oracle SMF Record

To interpret a Oracle SMF record, you first need to dump the SMF data set to a sequential data set. You can then write a program that does all of the following:

- Reads the sequential data set

- Selects only records with the Oracle record number
- Accesses the Oracle SMF record fields using the provided DSECT
- Prints the statistics

A sample program named ORAFMTO is provided in the SRCLIB library that you can customize for your installation. Refer to "[ORAFMT Sample Formatting Program](#)" on page D-8 for more information.

The Assembler copy file, ORASMFO, contains DSECTS that map and document the gateway SMF record fields. The ORASMFO data set member resides in the gateway SRCLIB library.

The ORASMFO file is divided into the following sections:

- The standard record header section, which contains offsets and lengths of the Net and accounting sections
- The correlation section
- The OSDI section
- The database engine section
- The OS/390 accounting section
- The Net section (if applicable), which contains information about the network origin of clients on a Oracle Net for OS/390 TCP/IP protocol network (IBM or SNS/TCPaccess)
- The OS/390 accounting section (if applicable)

Not all sections are present in all SMF records. For example, the OS/390 accounting section is present only in SMF records for batch and TSO users. When a section is present, the SMF record header contains the correct length for that section, which might be release dependent, and the count field, which contains 1. The length field for nonexistent sections contains 0.

Contents of the SMF Header Section

Table D-1 contains brief descriptions for the labels in the SMF header section. For a complete layout of the contents of the SMF header section, refer to the DSECT.

Table D-1 Contents of the SMF Header Section

ORASMF0 Label	Description
SMFHDR	Standard SMF header
SMFHLEN	Total length of the SMF record
SMFHSEG	Segment descriptor = 0
SMFHSSIN	SYS IND = X'80' Subsystem information to follow SYS IND = x'40' Subtype format record
SMFHREC	Record type default = 204 (decimal)
SMFHTIM	Timestamp, time (.01 seconds since midnight)
SMFHDAT	Timestamp, date (0ccyyydddf) c=0 for 19xx, c=1 for 20xx
SMFHSSYS	System id
SMFHSSI	OSDI Subsystem id
SMFHSSUB	Record subtype; 1 = accounting record
SMFSSRVC	OSDI service name
SMFSSSID	OSDI session id
SMFHHRV1	Reserved
SMFNETO	Offset to Net section
SMFACTO	Offset to OS/390 accounting section
SMFHHRV2	Reserved
SMFNETL	Length of Net section
SMFACTL	Length of OS/390 accounting section
SMFHHRV3	Reserved

Contents of the SMF Correlation Section

Table 9-2 contains brief descriptions for the labels in the SMF correlation section.

Table D–2 Contents of the SMF Correlation Section

ORASMF Label	Description
SMFAUTH	Authorization id = TSO logon id Batch userid on jobcard CICS USERID, TERM-ID, TRANS-ID, PROGRAM-ID, or OPID
SMFCORI	Correlation id = for TSO, logon id for batch, jobname for CICS, jobname NOt valid for Oracle Net
SMFCONN	Connection type (TSO, BATCH, CICS, VTAM, TCP/IP, IMS)
SMFASID	Users address space id (not valid for Oracle Net)
SMFOUSR	Gateway logon id
SMFTNAME	Originating terminal id (if available)
SMFPNAME	Originating program name (if available)
SMFGRPN	RACF group name (if available)
SMFJBID	JES job identifier
SMFENTRY	RDR jobcard entry date (batch and TSO only). This field is equivalent to the SMF5RST field in the SMF job termination (type 5) record.
SMFEDATE	RDR jobcard entry date (batch and TSO only). This field is equivalent to the SMF5RSD field in the SMF job termination (type 5) record.

Contents of the SMF OSDI Data Section

Table D–3 Contents of the SMF OSDI Data Section

ORASMF0 Label	Description
SMFTIM	Beginning timestamp, time (.01 second since midnight)
SMFDAT	Beginning timestamp, date (00yydddf), ending time and date in header
SMFDTAI	Data in
SMFDTAO	Data out

Table D-3 Contents of the SMF OSDI Data Section

ORASMF0 Label	Description
SMFXMCPU	Cross memory CPU time (TOD format)
SMFRPCS	RPC count
SMFHWST	High-water mark of storage used
SMFINV	Reason for invocation
SMFNORM	Normal termination
SMFabort	Clean up done

Contents of the SMF Database Engine Data Section

The database engine section is not applicable for Oracle Transparent Gateway for DB2.

Table D-4 Contents of the Database Engine Section

ORASMF Label	Description
SMFLRC	Logical read count
SMFPRC	Physical read count
SMFLWC	Logical writes
SMFDMC	DML COMMITs
SMFDMR	DML ROLLBACKs
SMFDED	DEADLOCKs
SMFHDLN	Length of SMF header

Contents of the SMF Oracle Net Data Section

Table D-5 Contents of the SMF Oracle Net Data Section

ORASMF Label	Description
SMFNET	Oracle Net section header
SMFNETL	Length of Net NIV information. The information contained in this section is specific to the Oracle Net driver in use. This information is variable length.
SMFNETA	Start of variable-length information

ORAFMT Sample Formatting Program

A sample program, ORAFMT, is provided with the gateway SMF interface to format gateway SMF records. ORAFMT is an Assembler program that reads and formats SMF accounting records with the default gateway type of 199. It reads records from a variable-blocked sequential data set and writes the formatted records to a fixed block sequential data set with a logical record length of 132.

The sample ORAFMT program is in the gateway SRCLIB library. The following members are included:

ORAFMTCL contains sample JCL to compile and link ORAFMT.
ORAFMTGO contains sample JCL to run ORAFMT.

ORAFMT extracts and prints the following values from the gateway SMF records:

Table D-6 SMF Record Values

Value	Description
SSN	OSDI Subsystem name
SERVICE	OSDI Service name
SMFAUTH	Oracle userid assigned to the session using the database link.
SMFCONN	Connection type (TSO, BATCH, VTAM, TCP/IP)
ORACLE ID	User id assigned to the database link when the link is created. If the dblink is created without an associated userid, it defaults to the Oracle userid.
DATE	Start date of Oracle session
TIME	Start time of Oracle session
CPU SECONDS	Total CPU seconds used in the gateway address space. This number does not include time spent in the Oracle server address space, nor in the DB2 address space.
LOG READS	Not applicable to the gateway
PHY READS	Not applicable to the gateway
LOG WRITES	Not applicable to the gateway
DMC	Not applicable to the gateway
DMR	Not applicable to the gateway
DED	Not applicable to the gateway
HI STG	High-water mark of main storage used by the session

If any of the values are too large for the precision of their column, then they are shown as a series of asterisks.

Sample output from the ORAFMT program:

SSN	SMFAUTH	SMFCONN	ORACLE ID	DATE	TIME	TOTAL CPU SEC	LOG READS	LOG WRITES	DMC	DMR	DDC	DDR	DED
ORA1	SSMITH	BATCH	SCOTT	95.070	13:46:17.74	.755	152	23	6	2	1	1	0
ORA1	SSMITH	BATCH	SCOTT	95.070	13:47:34.41	.193	88	23	6	2	1	1	0
ORA1	TJONES	BATCH	SYSTEM	95.070	13:48:06.27	.269	95	23	6	2	1	1	0
ORA1	SSMITH	BATCH	SCOTT	95.070	13:49:08.98	.084	36	10	3	2	0	0	0
ORA1	SSMITH	BATCH	SCOTT	95.070	13:50:19.60	.120	21	2	1	0	0	0	0
ORA1		BATCH	SCOTT	95.070	13:52:04.67	.105	10	16	2	2	0	0	0
ORA1		BATCH	SCOTT	95.070	13:52:36.61	1.028	115	9	3	1	0	0	0
ORA1		BATCH	SCOTT	95.070	13:52:42.47	.576	71	23	5	3	1	1	0
ORA1		BATCH	SYSTEM	95.070	13:54:14.83	.110	10	16	2	2	0	0	0

Data Dictionary Views

This appendix includes the gateway data dictionary views accessible to all users of an Oracle database server. Most views can be accessed by any user with SELECT privileges for DB2 catalog tables.

The following data dictionary views are included:

- [ALL_CATALOG](#) on page E-2
- [ALL_COL_COMMENTS](#) on page E-2
- [ALL_CON_COLUMNS](#) on page E-3
- [ALL_CONSTRAINTS](#) on page E-3
- [ALL_IND_COLUMNS](#) on page E-4
- [ALL_INDEXES](#) on page E-4
- [ALL_OBJECTS](#) on page E-5
- [ALL_SYNONYMS](#) on page E-6
- [ALL_TAB_COLUMNS](#) on page E-6
- [ALL_TAB_COMMENTS](#) on page E-7
- [ALL_TABLES](#) on page E-7
- [ALL_USERS](#) on page E-8
- [ALL_VIEWS](#) on page E-9
- [COLUMN_PRIVILEGES](#) on page E-9
- [OTGREGISTER](#) on page E-9
- [TABLE_PRIVILEGES](#) on page E-10

[USER_CATALOG](#) on page E-10

[USER_COL_COMMENTS](#) on page E-11

[USER_CONS_COLUMNS](#) on page E-11

[USER_CONSTRAINTS](#) on page E-11

[USER_INDEXES](#) on page E-12

[USER_OBJECTS](#) on page E-13

[USER_SYNONYMS](#) on page E-13

[USER_TAB_COLUMNS](#) on page E-14

[USER_TAB_COMMENTS](#) on page E-14

[USER_TABLES](#) on page E-15

[USER_USERS](#) on page E-16

[USER_VIEWS](#) on page E-16

If a dictionary item is described with N/A in the following tables, then the item is not available for the gateway. The following tables are all constructed with two columns. The first, or left, cell of each row contains the dictionary item name, and the second, or right, cell contains a description of the dictionary item.

ALL_CATALOG

All tables, views, synonyms, and sequences accessible to the user.

OWNER	Owner of the object
TABLE_NAME	Name of the object
TABLE_TYPE	Type of object

ALL_COL_COMMENTS

Comments on columns of accessible tables and views.

OWNER	Owner of the object
TABLE_NAME	Object name
COLUMN_NAME	Column name
COMMENTS	Comments on the column

ALL_CON_COLUMNS

Information about accessible columns in constraint definitions.

OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
TABLE_NAME	Name associated with the table containing the constraint definition
COLUMN_NAME	Name associated with the column specified in the constraint definition
POSITION	Original position of the column in the definition

ALL_CONSTRAINTS

Constraint definitions on accessible tables.

ALL_IND_COLUMNS

OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
CONSTRAINT_TYPE	Type of constraint definition
TABLE_NAME	Name associated with the table containing the constraint definition
SEARCH_CONDITION	Text of the search condition for the table check
R_OWNER	Owner of the table used in the referential constraint
R_CONSTRAINT_NAME	Name of the unique constraint definition for the referenced table
DELETE_RULE	Delete rule for the referential constraint
STATUS	Status of the constraint

ALL_IND_COLUMNS

Columns of the indexes on the accessible tables.

INDEX_OWNER	Owner of the index
INDEX_NAME	Name of the index
TABLE_OWNER	Table or cluster owner
TABLE_NAME	Table or cluster name
COLUMN_NAME	Column name
COLUMN_POSITION	Position of column within index
COLUMN_LENGTH	Indexed length of column

ALL_INDEXES

Description of indexes on tables accessible to the user.

OWNER	Owner of the index
INDEX_NAME	Name of the index

TABLE_OWNER	Owner of the indexed object
TABLE_NAME	Name of the indexed object
TABLE_TYPE	Type of indexed object
UNIQUENESS	Uniqueness status of the index
TABLESPACE_NAME	Name of the table space containing the index
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
PCT_FREE	N/A
BLEVEL	Depth of the index from its root block to its leaf blocks. A depth of one indicates the root block and the leaf block are the same.
LEAF_BLOCKS	Number of leaf blocks in the index
DISTINCT_KEYS	Number of distinct indexed values. For indexes enforcing UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table.
AVG_LEAF_BLOCKS_PE	N/A
AVG_DATA_BLOCKS_PE	N/A
CLUSTERING_FACTOR	N/A
STATUS	State of the index: VALID

ALL_OBJECTS

Objects accessible to the user.

ALL_SYNONYMS

OWNER	Owner of the object
OBJECT_NAME	Name of object
OBJECT_ID	Object number of the object
OBJECT_TYPE	Type of object
CREATED	N/A
LAST_DDL_TIME	N/A
TIMESTAMP	N/A
STATUS	State of the object

ALL_SYNONYMS

All synonyms accessible to the user.

OWNER	Owner of the synonym
SYNONYM_NAME	Name of the synonym
TABLE_OWNER	Owner of the object referenced by the synonym
TABLE_NAME	Name of the object referenced by the synonym
DB_LINK	N/A

ALL_TAB_COLUMNS

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

OWNER	Owner of the table or view
TABLE_NAME	Table or view name
COLUMN_NAME	Column name
DATA_TYPE	Data type of the column
DATA_LENGTH	Maximum length of the column in bytes
DATA_PRECISION	N/A
DATA_SCALE	Digits to the right of decimal point in a number
NULLABLE	Asks if 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.
COLUMN_ID	Sequence number of the column as created
DEFAULT_LENGTH	N/A
DATA_DEFAULT	N/A
NUM_DISTINCT	Number of distinct values in each column of the table
LOW_VALUE	Second lowest and second highest values for tables with more than three rows. These statistics are expressed in hexadecimal notation for the internal representation of the first 32 bytes of the values.
HIGH_VALUE	
DENSITY	N/A

ALL_TAB_COMMENTS

Comments on tables and views accessible to the user.

OWNER	Owner of the object
TABLE_NAME	Name of the object
TABLE_TYPE	Type of object
COMMENTS	Comments on the object

ALL_TABLES

Description of tables accessible to the user.

ALL_USERS

OWNER	Owner of the table
TABLE_NAME	Name of the table
TABLESPACE_NAME	Name of the table space containing the table
CLUSTER_NAME	N/A
PCT_FREE	N/A
PCT_USED	N/A
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
BACKED_UP	N/A
NUM_ROWS	Number of rows in the table
BLOCKS	N/A
EMPTY_BLOCKS	N/A
AVG_SPACE	N/A
CHAIN_CNT	N/A
AVG_ROW_LEN	Average length of a row in the table in bytes

ALL_USERS

Information about all users of the database.

USERNAME	Name of the user
USER_ID	N/A
CREATED	N/A

ALL_VIEWS

Text of views accessible to the user.

OWNER	Owner of the view
VIEW_NAME	Name of the view
TEXT_LENGTH	Length of the view text
TEXT	View text. Only the first row of text is returned, even if multiple rows exist.

COLUMN_PRIVILEGES

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

GRANTEE	Name of the user to whom access was granted
OWNER	User name of the object's owner
TABLE_NAME	Name of the object
COLUMN_NAME	Name of the column
GRANTOR	Name of the user who performed the grant
INSERT_PRIV	Permission to insert into the column
UPDATE_PRIV	Permission to update the column
REFERENCES_PRIV	Permission to reference the column
CREATED	Time stamp for the grant

OTGREGISTER

DB2 special registers.

CURRENT_SQLID	DB2 SQL authorization id
---------------	--------------------------

TABLE_PRIVILEGES

CURRENT_USER	DB2 primary authorization id
CURRENT_DATE	DB2 current date
CURRENT_TIME	DB2 current time
CURRENT_TIMESTAMP	DB2 current time stamp
CURRENT_TIMEZONE	DB2 current time zone
CURRENT_SERVER	DB2 location name of current server
CURRENT_PACKAGESET	DB2 collection id of the package used for running SQL statements

TABLE_PRIVILEGES

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

GRANTEE	Name of the user to whom access is granted
OWNER	Owner of the object
TABLE_NAME	Name of the object
GRANTOR	Name of the user who performed the grant
SELECT_PRIV	Permission to select from an object
INSERT_PRIV	Permission to insert into an object
DELETE_PRIV	Permission to delete from an object
UPDATE_PRIV	Permission to update an object
REFERENCES_PRIV	N/A
ALTER_PRIV	Permission to alter an object
INDEX_PRIV	Permission to create or drop an index on an object
CREATED	Time stamp for the grant

USER_CATALOG

Tables, views, synonyms, and sequences owned by the user.

TABLE_NAME	Name of the object
TABLE_TYPE	Type of object

USER_COL_COMMENTS

Comments on columns of user's tables and views.

TABLE_NAME	Object name
COLUMN_NAME	Column name
COMMENTS	Comments on the column

USER_CONS_COLUMNS

Information about columns in constraint definitions owned by the user.

OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
TABLE_NAME	Name associated with the table with the constraint definition
COLUMN_NAME	Name associated with the column specified in the constraint definition
POSITION	Original position of the column in the definition

USER_CONSTRAINTS

Constraint definitions on user's tables.

OWNER	Owner of the constraint definition
CONSTRAINT_NAME	Name associated with the constraint definition
CONSTRAINT_TYPE	Type of constraint definition
TABLE_NAME	Name associated with the table with the constraint definition
SEARCH_CONDITION	Text of the search condition for the table check
R_OWNER	Owner of the table used in the referential constraint
R_CONSTRAINT_NAME	Name of the unique constraint definition for the referenced table
DELETE_RULE	Delete rule for the referential constraint
STATUS	Status of the constraint

USER_INDEXES

Description of the user's own indexes.

INDEX_NAME	Name of the index
TABLE_OWNER	Owner of the indexed object
TABLE_NAME	Name of the indexed object
TABLE_TYPE	Type of the indexed object
UNIQUENESS	Uniqueness status of the index
TABLESPACE_NAME	Name of the table space containing the index
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
PCT_FREE	N/A

BLEVEL	Depth of the index from its root block to its leaf blocks. A depth of one indicates the root block and the leaf block are the same.
LEAF_BLOCKS	Number of leaf blocks in the index
DISTINCT_KEYS	Number of distinct indexed values. For indexes enforcing UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table.
AVG_LEAF_BLOCKS_PE	N/A
AVG_DATA_BLOCKS_PE	N/A
CLUSTERING_FACTOR	N/A
STATUS	State of the indexes: VALID

USER_OBJECTS

Objects owned by the user.

OWNER	Owner of the object
OBJECT_NAME	Name of object
OBJECT_ID	Object number of the object
OBJECT_TYPE	Type of object
CREATED	N/A
LAST_DDL_TIME	N/A
TIMESTAMP	N/A
STATUS	State of the object: VALID

USER_SYNONYMS

The user's private synonyms.

USER_TAB_COLUMNS

SYNONYM_NAME	Name of the synonym
TABLE_OWNER	Owner of the object referenced by the synonym
TABLE_NAME	Name of the object referenced by the synonym
DB_LINK	N/A

USER_TAB_COLUMNS

Columns of user's tables, views, and clusters.

TABLE_NAME	Table, view, or cluster name
COLUMN_NAME	Column name
DATA_TYPE	Data type of the column
DATA_LENGTH	Maximum length of the column in bytes
DATA_PRECISION	N/A
DATA_SCALE	Digits to the right of decimal point in a number
NULLABLE	Asks if 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.
COLUMN_ID	Sequence number of the column as created
DEFAULT_LENGTH	N/A
DATA_DEFAULT	N/A
NUM_DISTINCT	Number of distinct values in each column of the table
LOW_VALUE	Second lowest and second highest values for tables with more than three rows. These statistics are expressed in hexadecimal notation for the internal representation of the first 32 bytes of the values.
HIGH_VALUE	
DENSITY	N/A

USER_TAB_COMMENTS

Comments on the tables and views owned by the user.

TABLE_NAME	Name of the object
TABLE_TYPE	Type of object
COMMENTS	Comments on the object

USER_TABLES

Description of the user's own tables.

TABLE_NAME	Name of the table
TABLESPACE_NAME	Name of the table space containing the table
CLUSTER_NAME	N/A
PCT_FREE	N/A
PCT_USED	N/A
INI_TRANS	N/A
MAX_TRANS	N/A
INITIAL_EXTENT	N/A
NEXT_EXTENT	N/A
MIN_EXTENTS	N/A
MAX_EXTENTS	N/A
PCT_INCREASE	N/A
BACKED_UP	N/A
NUM_ROWS	Number of rows in the table
BLOCKS	N/A
EMPTY_BLOCKS	N/A
AVG_SPACE	N/A
CHAIN_CNT	N/A
AVG_ROW_LEN	Average length of a row in the table in bytes

USER_USERS

Information about the current user.

USERNAME	Name of the user
USER_ID	N/A
DEFAULT_TABLESPACE	N/A
TEMP_TABLESPACE	N/A
CREATED	N/A

USER_VIEWS

Text of views owned by the user.

VIEW_NAME	Name of the view
TEXT_LENGTH	Length of the view text
TEXT	First line of view text

Quick Reference to Oracle SQL Functions

This appendix lists the Oracle SQL functions.

Oracle SQL Functions

ABS	NLS_LOWER
ACOS	NLS_UPPER
ADD_MONTHS	NLSSORT
ASCII	POWER
ASIN	RAWTOHEX
ATAN	REPLACE
ATAN2	ROUND
CEIL	ROWIDTOCHAR
CHAR_TO_ROWID	RPAD
CHR	RTRIM
CONVERT	SIGN
COS	SIN
COSH	SINH
DECODE	SOUNDEX
DUMP	SQRT
EXP	STDDEV
FLOOR	SUBSTR

GREATEST	SUBSTRB
HEXTORAW	SYSDATE
INITCAP	TAN
INSTR	TANH
INSTRB	TO_CHAR
LAST_DAY	TO_DATE
LEAST	TO_LABEL
LENGTH	TO_MULTI_BYTE
LENGTHB	TO_NUMBER
LN	TO_SINGLE_BYTE
LOG	TRANSLATE
LOWER	TRUNC
LPAD	UID
LTRIM	UPPER
MOD	USER
MONTHS_BETWEEN	USERENV
NEW_TIME	VARIANCE
NEXT_DAY	VSIZE
NLS_INITCAP	

Sample Applications

This appendix contains sample applications that can be used with the gateway.

The following examples are included:

[DB2IND](#) on page G-1

[ORAIND](#) on page G-3

DB2IND

DB2IND is a sample DB2 stored procedure that inserts a row into a DB2 table. This procedure uses the SIMPLE WITH NULLS linkage convention.

```

/*****
/* This DB2 stored procedure uses indicator variables to insert null */
/* values for DNAME and LOC columns of DB2 user table SCOTT.DEPT.   */
/* 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  */
/* WITH NULLS.                                                      */
/*                                                                    */
/* 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                                                    */
/*  ('DB2IND', ' ', ' ', 'DB2IND', 'N', '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;
/*****
/* Declare C variables for SQL operations on the parameters. These */
/* are local variables to the C program which you must copy to and */
/* from the parameter list provided to the stored procedure.      */
/*****
EXEC SQL BEGIN DECLARE SECTION;
long dno;                /* input parm - DEPTNO */
char dname[15];         /* input parm - DNAME */
char locale[14];       /* input parm - LOC   */
struct INDICATORS {
    short int  i1;
    short int  i2;
    short int  i3;
    short int  o;
} indvar;                /* indicator variable structure */
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]);
/*****
/* Copy indicator variable values for the parameter list.        */
/*****
    memcpy(&indvar, (struct INDICATORS *) argv[6], sizeof(indvar));
/*****
/* Issue SQL INSERT to insert a row into SCOTT.DEPT              */
/*****
EXEC SQL INSERT INTO SCOTT.DEPT VALUES
(:dno:indvar.i1, :dname:indvar.i2, :locale:indvar.i3);

```

```

/*****
/* Copy SQLCODE to the output parameter list.          */
/*****
    *(int *) argv[4] = SQLCODE;
    indvar.o = 0;
/*****
/* Copy indicator variable values back to the output parameter list. */
/*****
    memcpy( (struct INDICATORS *) argv[6], &indvar, sizeof(indvar));
    }

```

ORAIND

ORAIND is a sample host program that calls a DB2 stored procedure (DB2IND) to insert a row into a DB2 table. Embedded PL/SQL is used to manipulate the indicator variables.

```

/*****
/* This sample ProC program calls DB2 stored procedure DB2IND to      */
/* insert null values into DB2 user table SCOTT.DEPT. This calling    */
/* program uses embedded PL/SQL to pass indicator variables in the    */
/* parameter list of the DB2 stored procedure call.                    */
/*****
#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];
    short        ind1;
    short        ind2;
    short        ind3;
    short        oind;
    int          x;
EXEC SQL END DECLARE SECTION;
EXEC SQL INCLUDE SQLCA;
main()
{
/*****
/* Setup Oracle userid and password          */
/*****

```

```
strcpy(username.arr, "SCOTT");          /* copy the username */
username.len = strlen(username.arr);
strcpy(password.arr, "TIGER");         /* copy the password */
password.len = strlen(password.arr);
EXEC SQL WHENEVER SQLERROR GOTO sqlerror;
/*****
/* Logon to Oracle */
*****/
EXEC SQL CONNECT :username IDENTIFIED BY :password;
printf("\nConnected to ORACLE as user: %s\n", username.arr);
/* Delete any existing rows from DB2 table */
EXEC SQL DELETE from SCOTT.DEPT@GTWLINK where LOC='INDVARS';
EXEC SQL COMMIT;
/*----- begin pl/sql block -----*/
/*****
/* Insert 5 rows into DB2 table SCOTT.DEPT by invoking DB2 stored */
/* procedure DB2IND. Use indicator variables to pass null values to */
/* the stored procedure. The DB2 stored procedure will perform the */
/* INSERT. */
/*
/* SCOTT.DEPT table is defined on DB2 as:
/*
/*   DEPTNO    INTEGER;
/*   DNAME     CHAR(14);
/*   LOC       VARCHAR(13);
/*
*****/
EXEC SQL EXECUTE
DECLARE
  buf  char(10);
BEGIN
for i in 1 .. 5 loop
  :dept_no:ind1 := 10 * i;
  :dept_name:ind2 := null;
  :location:ind3 := null;
  SYSPROC.DB2IND@GTWLINK
  (:dept_no:ind1, :dept_name:ind2, :location:ind3, :code:oind, buf);
end loop;
END;
END-EXEC;
/*----- end pl/sql block -----*/
/*****
/* Verify row insertion. Use indicator variables to check columns */
/* for null values. Update the column with a value if column is */
/* null. */
*****/
```

```

/*****
for (x = 10; x < 60; x = x + 10)
{
    EXEC SQL SELECT deptno, dname, loc into
        :dept_no:ind1, :dept_name:ind2, :location:ind3
    from SCOTT.DEPT@GTWLINK where deptno = :x;
    if ((ind2 == -1) && (ind3 == -1))
    {
        printf("\nAfter INSERT\n");
        printf("\ndeptno = %d, dname = NULL, loc = NULL\n", dept_no);
        EXEC SQL UPDATE SCOTT.DEPT@GTWLINK set dname = 'TESTING',
            loc = 'INDVARS' where deptno = :x;
        EXEC SQL COMMIT;
    }
    EXEC SQL SELECT deptno, dname, loc into
        :dept_no:ind1, :dept_name:ind2, :location:ind3
    from SCOTT.DEPT@GTWLINK where deptno = :x;
    printf("\nAfter UPDATE:\n");
    printf("\ndeptno = %d, dname = %s, loc = %s\n",
        dept_no, dept_name, location.arr);
}
/*****
/* Logoff from Oracle */
/*****
EXEC SQL COMMIT RELEASE;
printf("\n\nHave a good day\n\n");
exit(0);
sqlerror:
    printf("\n% .70s \n", sqlca.sqlerrm.sqlerrmc);
EXEC SQL WHENEVER SQLERROR CONTINUE;
EXEC SQL ROLLBACK RELEASE;
exit(1);
}

```

Installation Reference

This appendix documents additional installation information that is referenced in the installation sections. The following topics are included:

- [Choosing Data Set Name Qualifiers](#) on page H-2
- [Data Set Name Qualifier Rules](#) on page H-3
- [OR@INST Panel Options](#) on page H-4

Choosing Data Set Name Qualifiers

Step 1 of the primary Oracle9i for OS/390 installation setup and initialization process creates the first of several OS/390 data sets. Later in the installation, you can specify the high-level and second-level data set name qualifiers that are used for subsequently created data sets. Refer to [Appendix A, "Data Set Names and Space Allocations"](#), for a list of the data sets and their default space allocations.

Oracle Corporation recommends that you use the same qualifiers for all of the installation-related data sets. At this time, you need to choose and use the qualifiers that were selected in ["Setup and Initialization Steps"](#) on page 4-3.

While choosing qualifiers, remember the following requirements:

- You must choose unique qualifiers.

Using different qualifiers ensures that the products in the product set are maintained in separate libraries as required.

Attention:

1. Do not use the same qualifiers that you have used for any other Oracle for OS/390 product set that you have previously installed. If you do, the installation procedures will delete and reallocate your current Oracle libraries.
 2. Do not concatenate these libraries with any existing libraries that you are running for previously installed product sets.
-
-

- In most OS/390 systems, some preparation is required before creating data sets with a new high-level data set name qualifier.

If you intend to use a new high-level qualifier for your Oracle data sets, then you must define an ALIAS before running the job that loads the installation JCL. If in doubt, ask your OS/390 systems programmer for assistance.

Data Set Name Qualifier Rules

The following two rules concern the value that was entered for **Target Data Set Name Qualifiers** on panel OR@PRIM during the installation process:

- The high-level qualifier can be up to nine characters, including a period.
Once entered, the first 8 characters become the initial value for the **High Level Qualifier** on panel ORINIP00, described in "[Step 1: Modify Tape Unit and Library Index \(Panel ORINIP00\)](#)" on page 4-10.
- The second-level qualifier can be up to 17 characters, including periods.
Once entered, the 17 characters become the initial value for the **Second Level Qualifier** for item 2 on panel ORINIP00.

These two rules allow for Oracle data sets with high-level and second-level qualifiers, as in the following examples:

```
aaaaaaaa
aaaaaaaa.bbbbbbbb.ccccccc
aaaa.bbbbbbbb.ccccccc
aaaaaaaa.bb.cc.dd.ee.ff.gg
```

where the initial values on panel ORINIP00 are seen in these examples as:

High Level Qualifier	Second Level Qualifier
aaaaaaaa	
aaaaaaaa	bbbbbbbb.ccccccc
aaaa	bbbbbbbb.ccccccc
aaaaaaaa	bb.cc.dd.ee.ff.gg

OR@INST Panel Options

Options 1 through 5 on installation panel OR@INST are:

- Option 1 starts the Oracle product installation process by bringing up panel ORPRIM0.
- Option 2 generates an INSTLIB member based on the options that were chosen from the installation panels available through option 1.
This INSTLIB member contains a batch job stream that is used to populate the INSTLIB with a series of Oracle product installation jobs.
- Option 3 starts the Oracle database customization process by bringing up panel ORNEWDB.
This option is only valid for the Oracle9i for OS/390 product set and must not be selected for any other product set.
- Option 4 generates an INSTLIB member based on the options that were chosen from the customization panels available through Option 3.
This INSTLIB member contains a batch job stream that is used to populate the INSTLIB with a series of customization jobs for the new Oracle database that is being defined.
This option is valid only for the Oracle9i for OS/390 product set and must not be selected for any other product set.
- Option 5 starts the Transparent Gateway for DB2 customization process by bringing up panel ORNEWGD.
This option is only valid for the Oracle Transparent Gateway for dB2 product set and must not be selected for any other product set.
- Option 6 generates an INSTLIB member based on the options that were chosen from the customization panels available through Option 5.
This option is only valid for the Oracle Transparent Gateway for DB2 product set and must not be selected for any other product set.

Option 9 relates only to option 1.

Option 9 can be used to nullify all of the product and language selections that were made by the user on panels ORPRODS and ORLANG. These panels are available only through Option 1. Option 9 nullifies the selections, and a message appears in the upper-right corner of the OR@INST panel screen when the reset is complete.

Option 9 is not the only method available to change the selections. The following method also exists: While completing the Option 1 panels, you can individually reset (change to blank) any selections on the ORPRODS panel by returning to the panel and making the necessary changes line-by-line.

Regardless of the method that is chosen to accomplish the reset, once a reset of any or all selections is done, the user must proceed again through all the panels of Option 1 before choosing Option 2. Otherwise, the generate fails and gives an error message stating that no products or languages were selected.

A

- abends, 5-37
 - reporting the error, 10-8
- abnormal termination, 5-37
- ABS SQL function, F-1
- accounting information, D-1
- ACOS SQL function, F-1
- ADD_MONTHS SQL function, F-1
- additional software requirements
 - IBM DB2, 3-4
 - ISPF, 3-4
 - REXX, 3-4
- Alert Logs, OSDI, 11-3
- ALL_CATALOG view, E-2
- ALL_COL_COMMENTS view, E-2
- ALL_CON_COLUMNS view, E-3
- ALL_CONSTRAINTS view, E-3
- ALL_DB_LINKS data dictionary view, 8-3
- ALL_IND_COLUMNS view, E-4
- ALL_INDEXES view, E-4
- ALL_OBJECTS view, E-5
- ALL_SYNONYMS view, E-6
- ALL_TAB_COLUMNS view, E-6
- ALL_TAB_COMMENTS view, E-7
- ALL_TABLES view, E-7
- ALL_USERS view, E-8
- ALL_VIEWS view, E-9
- ALTER SERVICE, OSDI command, B-8
- ALTER SERVICEGROUP, OSDI command, B-4
- ALTER SESSION CLOSE DATABASE LINK
 - statement, 8-3
- ALTER, OSDI command, B-3
- ANO, 1-4
- APARs, 3-3
- application
 - migrating to OSDI, B-7
- application server support, 1-4
- applications, 1-6
- architecture
 - gateway, 1-8
- array
 - processing, 9-2
 - size, 9-3
- ASCII
 - SQL function, F-1
 - tables, known restrictions, 2-4
- ASIN SQL function, F-1
- ASO
 - checklist
 - for setting up ASO encryption, 6-13
 - for testing, 6-13
 - connect client and server, 6-14
 - encryption, 6-12
 - reset configuration parameters on server, 6-14
 - set parameters for client, 6-13
 - set parameters for server, 6-13
 - testing, 6-13
- ATAN SQL function, F-1
- ATAN2 SQL function, F-1
- AUTHID column, 9-5
- AUTHLOAD
 - library, 4-22, 4-23
- AUTHLOAD library
 - job ORIJC00, 4-14
- authorization
 - id, 7-5, 9-22, D-6
- authorizing the load library, 4-22

Auto Registration, gateway features, 2-2
AVG
 option, 9-20
 Oracle database server function, 9-20
 SQL function, 9-18

B

before starting the gateway, 4-22
bind variables, 9-13, 9-14, 9-17, 9-21
 known restrictions, 2-4
 restrictions, 2-5
BPXPRMxx member, 6-8

C

CAF
 CONNECT call, 7-8
 OPEN call, 7-8, 7-9
 OPEN connection to DB2, 7-5
Call Attach Facility See CAF
calls
 CAF CONNECT, 7-8
 CAF OPEN, 7-8, 7-9
 PL/SQL, 9-4
CANCEL THREAD command, 8-5
canceling threads, 8-5
catalog views for DB2, 9-22
CCSID
 with NLS, C-4
CEIL SQL function, F-1
CHAR FOR BIT DATA format, 9-18
CHAR_TO_ROWID SQL function, F-1
character
 columns, 9-11, 9-13
 converting a string, 9-12
 sets
 ASCII, 2-4
 compatibility, C-2
 EBCDIC, 2-4
 string operations, 9-11
character set name, 5-5
checklist
 configuration, 5-2
 post installation, 4-2

 post-configuration, 5-3
 post-installation, 4-2
 setting up ASO, 6-13
 setup and initialization, 4-2
 testing ASO, 6-13
CHECKSUM command, 1-4
CHR SQL function, F-1
clauses
 CONNECT BY, 9-20
 CONNECT TO, 7-4, 8-2, 8-3
 DELETE, 9-20
 FOR UPDATE, 9-19, 9-20
 INSERT, 9-20
 SELECT FOR INSERT, 9-20
 SELECT FOR UPDATE, 9-20
 SQL
 DELETE, 9-15
 INSERT, 9-15
 SELECT WHERE, 9-15
 UPDATE, 9-15
 TO_DATE, 9-16
 USING, 8-2
 VALUES, 9-16
 WHERE, 9-19
 WHERE CURRENT OF CURSOR, 9-17
CMDLOAD
 library, 5-36
CNV_LIT_FMT (sidENV variable), 5-27
collecting Oracle statistics, D-1
COLLID column, 9-7
COLUMN_PRIVILEGES view, E-9
columns
 AUTHID, 9-5
 character, 9-11, 9-13
 COLLID, 9-7
 date, 9-16
 DB2 DATE, 9-13, 9-14
 IBM DATE, 9-13
 LONG, 8-16, 9-11
 LUNAME, 9-5
 Oracle DATE, 9-14
 Oracle ROWID, 9-17
 PROCEDURE, 9-5
 RAW, 2-4
 result for SELECT, 9-20

- ROWID, 9-17
- command
 - OS/390 System
 - MODIFY (F), 6-11
 - STOP (P), 6-11
 - OSDI
 - ALTER, B-3
 - ALTER SERVICE, B-8
 - ALTER SERVICEGROUP, B-4
 - DEFINE, B-3
 - DEFINE SERVICE, 5-10, B-5
 - DEFINE SERVICEGROUP, B-3
 - DISPLAY, 6-12, B-12
 - DISPLAY LONG, 6-12
 - DRAIN, 6-12, B-13
 - keyword abbreviations, B-15
 - list of commands, B-11
 - RESUME, 6-12, B-13
 - SHOW, B-3
 - SHOW SERVICE, B-10
 - SHOW SERVICEGROUP, B-5
 - START, 5-10, 6-10, B-11
 - STOP, 6-11, B-14
 - OSDI DEFINE SERVICE, 5-29
 - start, 7-2
 - STOP, 7-2
- commands
 - CANCEL THREAD, 8-5
 - CREATE SNAPSHOT, 8-15
 - CREATE TABLE, 8-14
 - DELETE, 9-20
 - EXECUTE, 1-6
 - INSERT, 8-15
 - MPM
 - DISPLAY EXIT, 7-6
 - gateway operation, 7-2
 - OS/390
 - DUMP, 10-10
 - SLIP, 10-9
 - SETPROG, 4-22, 4-23
 - SETSSI, 4-22, 4-23
 - SMF
 - SETSMF, D-3
 - SQL, 8-2
 - SQL*Plus
 - COPY, 8-9, 8-10, 8-15
 - DELETE, 9-20
 - INSERT, 8-9
 - commit confirm protocol, 1-7
 - compatible SQL functions, 9-18
 - compensated SQL functions, 9-19
 - CONCAT SQL function, 9-18
 - configuration jobs
 - sidJC01 and sidJC02, 5-11
 - configuration steps, 5-3
 - CONNECT BY clause, 9-20
 - CONNECT TO
 - clause, 7-4, 8-2, 8-3
 - statement, 8-3
 - userid, 8-3
 - CONVERT SQL function, F-1
 - converting a character string, 9-12
 - COPY command, 8-9, 8-15
 - COPY SQL*Plus command, 8-9
 - copying data, 8-9
 - from the DB2 server, 8-14
 - from the Oracle database server, 8-9
 - COS SQL function, F-1
 - COSH SQL function, F-1
 - COUNT
 - options, 9-21
 - COUNT(*) SQL function, 9-18
 - COUNT(DISTINCT expression) SQL function, 9-18
 - CPU requirements, 3-2
 - CREATE DATABASE LINK statement, 8-2
 - CREATE SNAPSHOT command, 8-15
 - CREATE TABLE
 - command, 8-14
 - statement, 1-5
 - creating the ISPF libraries, 4-4
 - CSECT for entry point, 5-8
 - CSV210I message, job ORIJC00, 4-14
 - CURRDEGREE (sidENV variable), 5-23

D

- DASD
 - data set, 5-37
- data control language See DCL
- data definition language See DDL

- data dictionary, 5-7
 - SYSIBM.SYSPROCEDURES table, 9-7
 - table SYSIBM.SYSPROCEDURES, 9-7
 - using, 9-22
 - views, 8-3, 9-21, E-1
 - ALL_CATALOG, E-2
 - ALL_COL_COMMENTS, E-2
 - ALL_CON_COLUMNS, E-3
 - ALL_CONSTRAINTS, E-3
 - ALL_DB_LINKS, 8-3
 - ALL_IND_COLUMNS, E-4
 - ALL_INDEXES, E-4
 - ALL_OBJECTS, E-5
 - ALL_SYNONYMS, E-6
 - ALL_TAB_COLUMNS, E-6
 - ALL_TAB_COMMENTS, E-7
 - ALL_TABLES, E-7
 - ALL_USERS, E-8
 - ALL_VIEWS, E-9
 - COLUMN_PRIVILEGES, E-9
 - DBA_DB_LINKS, 8-3
 - OTGREGISTER, 9-22
 - TABLE_PRIVILEGES, E-10
 - USER_CATALOG, E-10
 - USER_COL_COMMENTS, E-11
 - USER_CONS_COLUMNS, E-11
 - USER_CONSTRAINTS, E-11
 - USER_DB_LINKS, 8-3
 - USER_INDEXES, E-12
 - USER_OBJECTS, E-13
 - USER_SYNONYMS, E-13
 - USER_TAB_COLUMNS, E-14
 - USER_TAB_COMMENTS, E-14
 - USER_TABLES, E-15
 - USER_USERS, E-16
 - USER_VIEWS, E-16
- data set names
 - choosing data set name qualifiers, H-2
 - on distribution media, 3-5
- data sets
 - DASD, 5-37
 - distribution media, 3-5
 - dump, 5-37
 - names, 3-5
 - partitioned, A-2
 - SYS1.DUMP, 10-10
 - SYSOUT, 5-37
 - system dump, 10-10
- data types
 - character string, 9-11
 - DATE, 9-12
 - DB2
 - CHAR, 9-17
 - GRAPHIC, 9-17
 - VARCHAR, 9-12, 9-17
 - IBM DATE, 9-13
 - LONG, 9-12
 - mapping, 9-10
 - numeric, 9-16
 - Oracle
 - database server, 9-13
 - DATE, 9-12
 - RAW, 9-17
 - restrictions, 9-10
 - TIME, 9-12
 - TIMESTAMP, 9-12
- database
 - accounting information, D-1
 - triggers, 1-4
- database link, 8-2
 - accessing data, 8-3
 - behavior, 8-1
 - creating, 8-2
 - data dictionary views
 - ALL_DB_LINKS, 8-3
 - DBA_DB_LINKS, 8-3
 - USER_DB_LINKS, 8-3
 - dropping links, 8-3
 - examining, 8-3
 - guidelines, 8-3
 - limits, 8-4
 - using Oracle Net, 8-2
- DATE
 - data type, 9-12
 - DB2 columns, 9-13, 9-14
- date
 - arithmetic, 9-16
 - columns, 9-16
 - considerations, 9-14
 - DATE data type, 9-12, 9-13

- DB2
 - DATE columns, 9-13, 9-14
 - local date exit, 9-13, 9-14
- INSERT
 - c, 9-15
- LOCAL DB2 data format, 9-13
- NLS_DATE_FORMAT parameter, 9-15
- operations, 9-12
- TIME data type, 9-12
- TIMESTAMP data type, 9-12
- TO_DATE function, 9-14, 9-15, 9-16

DB2

- ASCII data to EBCDIC, 9-11
- bind
 - JCL, 9-7
- CAFOPEN connection, 7-5
- catalog, 9-12
- CHAR data type, 9-17
- CICS, 9-6
- collection id, 9-7
- connection exit, 7-8
- CURRENT SERVER value, 9-5
- cursors, 8-5
- data types
 - CHAR, 9-17
 - VARCHAR, 9-17
- database
 - read, 9-2
 - write, 9-2
- DATE columns, 9-13, 9-14
- differences from Oracle database server, 9-20
- GRAPHIC data types, 9-17
- IMS, 9-6
- known restrictions
 - for plan, 2-4
- load library, 5-6
- local date exit, 9-13, 9-14, 9-15
- LOCAL date format, 9-13
- native
 - SQL, 1-5
 - stored procedures, 1-6
- optional security class, 5-7
- plan name, 5-6
- PL/SQL, 9-7
- PROC1, 9-5
- security
 - checking, 7-9
 - profile, 7-5
- server triggers, 8-9
- special registers for authorization id, 9-22, E-9
- SQL, 9-7
- SQL statements, 9-8, 9-14
- statement CREATE TABLE, 1-5
- stored procedures, 8-6, 9-4, 9-6
- subsystem name, 5-6
- tables, 8-16, 9-12
- VARCHAR data type, 9-12, 9-17

DB2 gateway plan binding

- binding the DB2 gateway plan, 5-15

DB2 package

- binding of, 5-12

DB2 Server, 1-8

DB2 Universal Database (UDB), 1-8

DB2AUTH, 5-8

- exit, 7-9
- for DB2 connection exit, 5-8
- member, 7-9
- security considerations, 7-5

DB2CAP (sidENV variable), 5-25

DB2DESCTAB (sidENV variable), 5-23

DB2I bind jpackage panel, 5-13

DB2IND sample DB2 stored procedure, G-1

DB2LONGMSG (sidENV variable), 5-23

DB2PLAN (sidENV variable), 5-25

DB2READONLY

- parameter, 8-6

DB2READONLY (sidENV variable), 5-24

DB2SSN (sidENV variable), 5-25

DB2STATS (sidENV variable), 5-24

DB2WARNING (sidENV variable), 5-23

DBA_DB_LINKS data dictionary view, 8-3

DBCS support, 9-17

DBMS_HS_PASSTHROUGH.EXECUTE_

- IMMEDIATE function, 9-8

DCL

DD basic tables, known restrictions, 2-4

DD statement

- NET8LOG, 6-7
- STEPLIB, 6-7

DD statements, 5-32

- ORA\$ENV, C-3
- SYSDUMP, 5-37
- DDL statement, 9-8
- debugging messages, 5-7
- DECODE SQL function, F-1
- DEDICATED_TCB, 5-18
- DEDTCB, 5-18
- default
 - record types for SMF, D-3
- DEFINE command
 - supported definition command, B-3
- DEFINE SERVICE, OSDI command, B-5
- DEFINE SERVICEGROUP, OSDI command, B-3
- definition
 - gateway service, 5-29
- definition commands, B-3
- DELETE
 - clause, 9-20
 - command, 9-20
 - SQL clause, 9-15
- DFLTLIST member, A-2
- dictionary mapping, 1-4
- disk
 - space requirements, 3-2
- disk unit specification, 4-6
- DISPLAY LONG, OSDI command, 6-12
- DISPLAY, OSDI command
 - defined, B-12
 - with Oracle Net service, 6-12
- distributed
 - DB2 transactions, 8-8
 - queries, 8-7, 8-8
- distribution
 - media
 - data set names, 3-5
 - tape, 3-5
- documentation
 - errors, 10-7
- DRAIN, OSDI command
 - defined, B-13
 - preventing network connections, 6-12
- DROP DATABASE LINK statement, 8-3
- DSN0, default DB2 subsystem name, 5-6
- DSN3@ATH exit, 5-8
- DSN510 high-level qualifier for DB2 load

- libraries, 5-6
- DSNEXIT load library, 5-6
- DSNR security class, 5-7
- DSNXVDTX, 5-8
- DSNXVDTX DB2 date exit information, 5-8
- DSPRT library, A-2
- dump data sets, 5-37
- DUMP SQL functions, F-1

E

- EBCDIC tables, known restrictions, 2-4
- EMPTYSTR_TO_NULL_OFF (sidENV variable), 5-27
- enclave SRB
 - Oracle Net architecture, 6-3
- encryption, ASO, 6-12
- environment
 - heterogeneous, 8-8
 - operational, 10-6
 - security, 7-5
 - UNIX, 8-4
 - variables
 - DB2READONLY parameter, 8-6
- EPLPA, moving modules into, 5-36
- error
 - 12660, 6-12
 - JCL error associated with missing libraries, 4-14
 - message traffic to the console, 6-11
 - messages for network region JCL, 6-7
 - ORA-12660, 6-14
 - ORA-2063, example message, 10-3
- errors
 - diagnosing, 10-5, 10-7
 - incorrect output, 10-8
 - interpreting, 10-3
 - mapping, 10-2
 - messages, 10-1, 10-2
 - reporting
 - abends, 10-8
 - documentation, 10-5, 10-6, 10-7
 - GTF, 10-11
 - incorrect output, 10-8
 - Oracle database server error messages, 10-8
 - performance, 10-9

- program check, 10-8
- program loop, 10-9
- sending tapes, 10-5
- system dump data sets, 10-10
- system dumps, 10-10
- stored procedure, 10-4

EXECUTE

- command, 1-6

exits

- DB2
 - connection, 7-8
 - local date, 9-13, 9-14, 9-15
- DB2AUTH, 7-9
- G4AUTH, 7-9
- gateway local date, 9-15
- modules, 7-6
- sample programs, 7-9
- user, 7-5

EXP SQL function, F-1

extended pageable link pack area See EPLPA

F

FDS_CLASS_VERSION (sidENV variable), 5-24

features

- 9.0.1 gateway, 2-2
- Oracle snapshot, 8-8
- synonym, 8-6

fetch reblocking, 9-3

fields

- PARMLIST, 9-6
- SMFWCPU, 8-5

filenames

- OSDI listener, 6-3

files

- SPOOL, 5-37
- SQLNET.ORA, 8-5
- VSAM, 9-7

FLOOR SQL function, F-1

FLUSH_CACHE_ON_COMMIT (sidENV variable), 5-25

FOR BIT DATA option, 2-4, 9-11

FOR UPDATE clause, 9-19, 9-20

- NOWAIT option, 9-20

FPS

- sidFPS PARMLIB library member, 5-10

functions

- COUNT, 9-21
- DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE, 9-8
- KEEPALIVE, 8-4
- MIN, 9-20

Oracle database server

- AVG, 9-20
- MAX, 9-20
- MIN, 9-20
- SUM, 9-20
- TO_DATE, 9-20

ROWID, 9-17

SQL, 9-18

- ABS, F-1
- ACOS, F-1
- ADD_MONTHS, F-1
- advanced, 9-19
- ASCII, F-1
- ASIN, F-1
- ATAN, F-1
- ATAN2, F-1
- AVG, 9-18
- CEIL, F-1
- CHAR_TO_ROWID, F-1
- CHR, F-1
- compatible, 9-18
- compensated, 9-19
- CONCAT, 9-18
- CONVERT, F-1
- COS, F-1
- COSH, F-1
- COUNT(*), 9-18
- COUNT(DISTINCT expression), 9-18
- DECODE, F-1
- DUMP, F-1
- EXP, F-1
- FLOOR, F-1
- GREATEST, F-2
- HEXTORAW, F-2
- 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
MAX, 9-18
MIN, 9-18
MOD, F-2
MONTHS_BETWEEN, F-2
NEW_TIME, F-2
NEXT_DAY, F-2
NLS_INITCAP, F-2
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
SINH, F-1
SOUNDEX, F-1
SQRT, F-1
STDDEV, F-1
SUBSTR, F-1
SUBSTRB, F-2
SUM, 9-18
SYSDATE, F-2
TAN, F-2
TANH, 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, F-2
translated, 9-18

TRUNC, F-2
UID, F-2
UPPER, F-2
USER, F-2
USERENV, F-2
VARIANCE, F-2
VSIZE, F-2
SUM, 9-20
TO_DATE, 9-14, 9-15, 9-16

G

G4AUTH
 exit, 7-9
 for DB2 logon exit, 5-7
 sample user exit, 7-5
G4DB2PLN
 DB2 plan name, 5-6
G4sid, 5-31
gateway
 advantages
 migration and coexistence, 1-7
 multisite transactions, 1-7
 security, 1-8
 site autonomy, 1-7
 two-phase commit, 1-7
 architecture, 1-8
 components, 1-8
 DB2 server, 1-8
 Oracle database server, 1-8
 configuration, C-3
 CPU time, 8-5
 development applications, 9-2
 how it works, 1-9
 local date exit, 9-15
 messages, 10-4
 MPM, 7-2
 protection of current investment, 1-8
 security, 7-4, 7-5
 authorization id, 7-5
 DB2AUTH exit, 7-8
 Oracle user exit facility, 7-5
 SAF router, 7-5
 sample exit programs, 7-9
 secondary authorization id, 7-5

- specifying an exit module, 7-6
- gateway configuration, 5-3
- gateway messages, format explained, 10-3
- gateway region parameters (sidPARM), 5-18
- gateway service
 - opening a, 11-5
- gateway service DEFINE parameters, 5-29
- gateway service definition, 5-29
- gateway service, configuring a, 11-4
- GETHOSTBYNAME API, Oracle Net IBM TCP/IP HPNS, 6-8
- GREATEST SQL function, F-2
- GTF, 10-11

H

- heterogeneous replication
 - STREAMS, 2-3
- HEXTORAW SQL function, F-2
- high-level qualifier, 4-22
 - choosing data set name qualifiers, H-2
 - data set name qualifier rules, H-3
 - INDEX parameter, 4-6
- HPNS interface, 11-7
- HS initialization parameters, 5-28
- HS_OPEN_CURSORS
 - parameter, 8-5
- HS_RPC_FETCH_REBLOCKING
 - parameter, 9-3
- HS_RPC_FETCH_SIZE
 - initialization parameter, 3-2
 - parameter, 9-3

I

- IBM DATE
 - column, 9-13
 - data type, 9-13
- IBM maintenance, 3-3
- IEAAPFxx member, 4-22
- IEASYSnn member, 4-22
- IEBUPDTE utility, 4-22
- incompatibilities, release, 11-7
- incorrect output errors, 10-8
- INIT_SESSIONS, 5-19

- INIT_STACK_SIZE, 5-19
- INITCAP SQL function, F-2
- initialization
 - parameters
 - HS_RPC_FETCH_SIZE, 3-2
- INITSESS, 5-19
- INSERT
 - clause, 9-20
 - command, 8-15
 - SQL clause, 9-15
 - statement, 10-5
- installation
 - before starting the gateway, 4-22
 - checking distribution kit, 3-5
 - jobs, 4-13
 - post installation steps, 5-36
- INSTR SQL function, F-2
- INSTRB SQL function, F-2
- interface, Oracle array processing, 9-2
- Internet support, 1-4
- Intranet support, 1-4
- INTSTKSZ, 5-19
- IPL
 - avoiding IPL after job ORIJC00, 4-14
- IPLing OS/390, 4-14, 4-22
- isolation level, 5-17
- ISPCLIB
 - library, A-2
- ISPF
 - EDIT, 4-22
- ISPF profile files
 - selecting, 5-3
- ISPSLIB
 - library, A-2
- ISPTLIB library, A-2
- IXCF support, 11-7

J

- JCL
 - DB2 bind, 9-7
 - define an ALIAS before loading the installation
 - JCL, H-2
 - errors associated with missing libraries, 4-14
 - loading Oracle installation JCL, 4-3

- MPM startup, 5-37
- ORAFMT sample, D-10
 - running generated installation jobs, 4-13
 - suffix for Oracle JCL procedures, 4-12
- JCL, gateway instance, 11-4
- JCL, network region, 6-6
- job
 - job to load the installation JCL, 4-3
 - ORDJA01, 5-10
 - ORIJB00, 4-14
 - ORIJC00, 4-14
 - OSPIJA00 job parameters, 4-6
- JOBNAME (gateway service DEFINE parameter), 5-31
- JOBNAME parameter, 4-11, 5-9
- jobs
 - sidJC02, 5-7
- JOIN capability, 1-3

K

- Katakana DB2 support, 9-17
- KEEPALIVE
 - function, 8-4, 8-5
 - Oracle Net, 8-5
- keyword abbreviations, OSDI command, B-15
- known problems, 2-3
- known restrictions, 2-4
 - ASCII and EBCDIC character sets, 2-4
 - binary literal notation, 2-4
 - DB2
 - plan, 2-4
 - DD basic tables and views, 2-4
 - literal values, 2-4
 - OCI, 2-4
 - Oracle database server SQL, 2-5
 - precompiling, 2-5
 - RAW data, 2-5
 - savepoints, 2-6
 - SQL, 2-5

L

- Language Environment runtime library (LE)
 - Oracle Net, 6-8

- LAST_DAY SQL function, F-2
- LDAP.ORA file, sample, 6-10
- LEAST SQL function, F-2
- LENGTH SQL function, F-2
- LENGTHB SQL function, F-2
- libraries
 - AUTHLOAD, 4-22, 4-23
 - authorized, 4-22
 - CMDLOAD, 5-36
 - DSNEXIT, 5-6
 - DSPT, A-2
 - ISPCLIB, A-2
 - ISPSLIB, A-2
 - ISPTLIB, A-2
 - load, 4-22
 - PARMLIB, 8-5
 - SRCLIB, 7-9
 - SYS1.PARMLIB, 4-22, 4-23
- link pack area See LPA
- linkage conventions
 - SIMPLE, 9-7
 - SIMPLE WITH NULLS, 9-7
- linklist library, system
 - job ORIJC00, 4-14
- listener
 - OSDI, See OSDI listener
- listener architecture, 6-2
- LN SQL function, F-2
- load library, authorizing, 4-22
- LOCAL DB2 date format, 9-13
- LOG SQL function, F-2
- LOGDATA
 - structure, 7-9
- LOGON_AUTH
 - LGNAUTH, 5-19
- LONG
 - column, 8-16, 9-11
 - data type, 9-12
- LOWER SQL function, F-2
- low-level qualifier, 4-22
- LPA, moving modules into, 5-36
- LPAD SQL function, F-2
- LTRIM SQL function, F-2
- LUNAME column, 9-5

M

MACRO API interface, with Oracle Net, 6-8

managing threads, 8-4

mapping error messages, 10-2

MAX

option, 9-20

Oracle database server function, 9-20

SQL function, 9-18

MAX_SESSIONS, 5-19

MAXFILEPROC parameter, 6-8

MAXSESS, 5-19

MAXSOCKETS parameter, 6-8

member

BXPFRMxx, 6-8

OSPIJA00, loads ISPF libraries, 4-4

members

DB2AUTH, 7-9

DFLTLIST, A-2

IEAAPFxx, 4-22

IEASYSn, 4-22

ORAFMTCL, D-8

ORAFMTGO, D-8

PARMLIB, 7-6

PROGxx, 4-22

SHIPLIST, A-2

sidENV, 8-5

SMFPRMxx, D-2

message

CSV210I, job ORIJC00, 4-14

messages, 10-2

debugging, 5-7

error, 10-1

format, 10-3

gateway, 10-2

generated by gateway, 10-4

mapped, 10-2

ORA-2025, 8-9

ORA-2070, 2-6

ORA-9100, 10-2

Oracle database server, 10-8

stored procedure, 10-4

migrating from MPM

cautions, 11-4

MIN

option, 9-20

Oracle database server function, 9-20

SQL function, 9-18

missing functionality, 10-10

Mobile Agents, 1-4

MOD SQL function, F-2

MODIFY (F), OS/390 command

communicating with a running Oracle Net service, 6-11

MONTHS_BETWEEN SQL function, F-2

MPM

operation, 7-2

startup JCL, 5-37

STEPLIB, 7-6

SYSABEND for startup, 5-37

SYSUDUMP for startup, 5-37

MPM commands

DISPLAY EXIT, 7-6

gateway operation, 7-2

MPM messages, obsolescence of, 11-3

MPM parameters

gateway subsystem, 7-2

MPM to OSDI migration, 11-7

MPM, compared to OSDI, 11-2

MPM/TNS coexistence, 11-10

multiple gateway services, configuring, 11-9

N

Net service

See OSDI listener

NET8LOG DD statement, 6-7

network service

configuring, 11-6

operating, 11-6

OSDI, 11-2

NEW_TIME SQL function, F-2

NEXT_DAY SQL function, F-2

NLS

DB2 configuration, C-4

NLS_DATE_FORMAT

parameter, 9-15

NLS_DATE_FORMAT (sidENV variable), 5-26

NLS_INITCAP SQL function, F-2

NLS_LANG

- parameter, C-3
- specification, C-2
- NLS_LANG (sidENV variable), 5-25
- NLS_LOWER SQL function, F-1
- NLS_NCHAR (sidENV variable), 5-26
- NLS_UPPER SQL function, F-1
- NLSSORT SQL function, F-1
- NOTIFY parameter, 4-11, 5-9
- NOWAIT option, 9-20
- numeric data types, 9-16

O

- OCI, 2-4, 9-7, 10-3
- OPEN_LINKS parameter, 8-4
- operator initiated dumps, 10-11
- options
 - AVG, 9-20
 - FOR BIT DATA, 9-11
 - MAX, 9-20
 - MIN, 9-20
 - NOWAIT, 9-20
 - SUM, 9-20
 - TO_DATE, 9-20
- OR@INST panel, 4-8, 5-4, H-4
- OR@PRIM panel, 4-8, 5-4, H-3
- ORA\$ENV (DD statement), 5-32
- ORA\$ENV DD statement, C-3
- ORA_MAX_DATE (sidENV variable), 5-26
- ORA-12660 error, 6-14
- ORA-2025 message, 8-9
- ORA-2063, example error message, 10-3
- ORA-2070 message, 2-6
- ORA-9100 message, 10-2
- Oracle Advanced Security Option
 - See ASO
- Oracle Call Interface See OCI
- Oracle database server
 - data type, 9-13
 - DATE columns, 9-14
 - dump data sets, 5-37
 - functions
 - AVG, 9-20
 - MAX, 9-20
 - MIN, 9-20
 - SUM, 9-20
 - TO_DATE, 9-20
- subsystem
 - accounting information, D-1
 - error reporting, 10-8
 - resource usage, D-1
 - statistics, D-1
 - triggers, 8-9
 - user exit facility, 7-5
- Oracle Developer, 1-6
- Oracle Discoverer, 1-6
- Oracle distribution libraries
 - job ORIJB00, 4-14
- Oracle libraries
 - creating, 4-4
- Oracle Net
 - IBM TCP/IP
 - requirements, 3-5
 - KEEPALIVE function, 8-5
 - OSDI listener, 6-2
 - overview, 6-2
- Oracle net
 - requirements
 - resource, 3-2
- Oracle Primary Option Menu panel, 4-13
- Oracle Support Services, 10-5
- ORACLE2PC table, 2-4, 8-8
- Oracle9i database server
 - database triggers, 1-4
 - distributed query optimization, 1-3
 - stored procedures, 1-4
 - two-phase commit protection, 1-4
- ORAFMT program for SMF, D-4, D-8
- ORAFMTCL
 - member, D-8
 - sample JCL, D-8
- ORAFMTGO
 - member, D-8
 - sample JCL, D-8
- ORAIND sample DB2 stored procedure, G-3
- ORARECID (sidENV variable), 5-25
- ORDBIP90 panel, 5-9
- ORDJA01 job, 5-10
- ORDSN panel, 4-12
- ORDSNO panel, 4-12

- ORGDIP10 panel, 5-5, 5-10
- ORGDIP20 panel, 5-6, 5-7
- ORGDIP35 panel, 5-8
- ORGDIP40 panel, 5-8
- ORGDIP50 panel, 5-9
- ORGDJA01 panel, 5-10
- ORIJB00 job, 4-14
- ORIJC00 job, 4-14
- ORINIP00 panel, 4-10, H-3
- ORINIP15 panel, 4-11
- ORINIP20 panel, 4-11
- ORINIP25 panel, 4-12
- ORINIP60 panel, 4-13
- ORLANG panel, 4-11
- ORNEWDB panel, 5-5, H-4
- ORPRIM0 panel, 4-10, H-4
- ORPRODS panel, 4-10
- OS/390
 - commands
 - DUMP, 10-10
 - SLIP, 10-9
 - link pack areas, 5-36
 - operating system, supported releases, 3-3
 - operating system, supported versions, 3-2
- OS/390 UNIX System Services (USS)
 - with network service, 6-8
- OSDI
 - command
 - keyword abbreviations, B-15
 - DEFINE SERVICE command, 5-10
 - listener, 6-2
 - START command, 5-10
 - subsystems, 11-2
 - upgrade or migration to, 11-8
- OSDI command verbs, 7-3
- OSDI commands
 - access to, 7-2
- OSDI DEDICATED_TCB, 5-18
- OSDI DEFINE SERVICE, 5-29
- OSDI listener
 - configuration, 6-4
 - DEFINE parameters, 6-5
- OSDI listener architecture, 6-2
- OSDI network service, 11-2
- OSDSI

- differences from MPM, 11-2
- OSPIJA00 job, parameters, 4-6
- OSPIJA00 member
 - member that loads ISPF libraries, 4-4
 - symbolic parameters in, 4-5
- OTGDB2 userid, 5-6
- OTGREGISTER view, 9-22, E-9

P

- pageable link pack area See PLPA
- panel
 - OR@INST, 4-8, 5-4, H-4
 - OR@PRIM, 4-8, 5-4, H-3
 - Oracle Primary Option Menu, 4-13
 - ORDBIP90, 5-9
 - ORDSN, 4-12
 - ORDSNO, 4-12
 - ORGDIP10, 5-5, 5-10
 - ORINIP00, 4-10, H-3
 - ORINIP15, 4-11
 - ORINIP20), 4-11
 - ORINIP25, 4-12
 - ORINIP60, 4-13
 - ORLANG, 4-11
 - ORNEWDB, 5-5, H-4
 - ORPRIM0, 4-10, H-4
 - ORPRODS, 4-10
- panel ORGDIP20, 5-6, 5-7
- panel ORGDIP35, 5-8
- panel ORGDIP40, 5-8
- panel ORGDIP50, 5-9
- panel ORGDJA01, 5-10
- parameter
 - ASO, setting for server, 6-13
 - configuration, resetting on server, 6-14
 - encryption, 6-13
 - job OSPIJA00, 4-6
 - JOBNAME, 4-11, 5-9
 - MAXFILEPROC, 6-8
 - MAXSOCKETS, 6-8
 - NOTIFY, 4-11, 5-9
 - REGION
 - network region JCL, 6-7
 - setting, for client, 6-13

parameters

- DB2READONLY, 8-6
- HS_OPEN_CURSORS, 8-5
- HS_RPC_FETCH_REBLOCKING, 9-3
- HS_RPC_FETCH_SIZE, 9-3
- initialization
 - HS_RPC_FETCH_SIZE, 3-2
- list for stored procedures, 9-6
- MPM
 - startup, D-2
- NLS_DATE_FORMAT, 9-15
- NLS_LANG, C-3
- OPEN_LINKS, 8-4
- Oracle Net TNSNAMES SID, 2-3
- READONLY, 8-6
- SQLNET.EXPIRE_TIME, 8-5
- stored procedures list, 9-6
- PARM (gateway service DEFINE parameter), 5-31
- PARMLIB
 - library, 8-5
 - member, 7-6
- PARMLIB member
 - editing of, 5-18
- PARMLIB members
 - DEDICATED_TCB, 5-18
 - INIT_SESSIONS, 5-19
 - INIT_STACK_SIZE, 5-19
 - LOGON_AUTH, 5-19
 - MAX_SESSIONS, 5-19
 - SERVER_LOADMOD, 5-18
 - SMF_STAT_RECNO, 5-20
 - TRACE_DSNAME, 5-20
- PARMLIST field, 9-6
- passthrough, 1-5, 8-6, 9-7, 9-8, 9-9
 - result sets, 9-8, 9-9
- passthru See passthrough
- PDS, A-2
- performance
 - issues, 10-9
- PLPA, 5-37
- PLPA, moving modules into
- PL/SQL, 9-7
 - call, 9-4
 - DB2, 9-7
 - routine, 1-6

- standard Oracle, 1-6
- POWER SQL function, F-1
- Precompiler limitations, 2-5
- PROC (gateway service DEFINE parameter), 5-31
- PROCEDURE column, 9-5
- procedures
 - error example for stored procedures, 10-4
 - native stored, 1-6
 - sample DB2 DB2IND, G-1
 - sample for DB2 ORAIND, G-3
 - stored, 1-4, 1-5, 8-6, 9-4
- processing SQL, 9-19
- product set, 2-2
- program loop, 10-9
- PROGxx member, 4-22
- protection of investment, 1-8
- protocols
 - commit confirm, 1-7
- PTFs, 3-3

Q

- queries, distributed, 8-7

R

- RACF
 - with network service, 6-8
- RAW
 - columns, known restrictions, 2-4
 - data type, 9-17
 - data, known restrictions, 2-5
 - format, 9-18
- RAWTOHEX SQL function, F-1
- RECO user ID, 5-5
- recording SMF information, D-2
- reentrant modules, moving to EPLPA
- REGION parameter, 6-7
- REPLACE SQL function, F-1
- replication, 8-8
- requirements
 - CPU, 3-2
 - disk space, 3-2
 - IBM DB2, 3-4
 - ISMF, 3-4

- Oracle Net, 3-2, 3-4
- REXX, 3-4
- virtual memory, 3-2
- resource usage, D-1
- RESUME command
 - defined, B-13
 - Oracle Net, 6-12
- RMODE linkage, 5-36
- ROUND SQL function, F-1
- routines
 - PL/SQL, 1-6
- ROWID
 - column, 9-17
 - DELETE, 9-17
 - restrictions, 9-17
 - snapshots, 9-17
 - UPDATE, 9-17
 - WHERE CURRENT OF CURSOR
 - clause, 9-17
 - function, 2-5, 9-17
- ROWIDTOCHAR SQL function, F-1
- RPAD SQL function, F-1
- RTRIM SQL functions, F-1

S

- SAF
 - calls, 5-7
 - considerations, 7-5
 - router, 7-5
- savepoints, known restrictions, 2-6
- scripts
 - SQL DD, 2-4
- secondary authorization id, 7-5
- second-level qualifier, data set name qualifier
 - rules, H-3
- security, 5-7
 - advanced, 1-4
 - ANO, 1-4
 - class, 5-7
 - encryption, 1-4
 - for gateway, 7-4, 7-5
 - SAF router considerations, 7-5
 - specifying an exit module, 7-6
 - user exit, 7-5

- Security Authorization Facility See SAF
- SELECT
 - result column list, 9-20
 - statement, 8-6, 9-3, 9-19, 9-20
- SELECT FOR DELETE clause, 9-20
- SELECT FOR INSERT clause, 9-20
- SELECT FOR UPDATE clause, 9-20
- SELECT WHERE SQL clause, 9-15
- SELECT_CONCAT_ON (sidENV variable), 5-28
- SERVER_LOADMOD, 5-18
- service
 - definition commands, B-5
 - service group
 - definition commands, B-3
 - service group, definition of, 11-3
 - service name
 - gateway (DEFINE parameter), 5-30
 - session connection, 8-1
 - SETPROG system command, 4-22, 4-23
 - SETSSI system command, 4-22, 4-23
 - setting of, 5-17
 - SHIPLIST member, A-2
 - SHOW SERVICE, OSDI command, B-10
 - SHOW SERVICEGROUP, OSDI command, B-5
 - SHOW, OSDI command, B-3
- sid
 - gateway instance, 11-4
- SID (gateway service DEFINE parameter), 5-31
- sidENV
 - HS initialization parameters, 5-28
- sidENV member, 8-5
- sidENV parameters, 5-21
- sidJCO0 job, 5-10
- sidJCO1 and sidJCO1
 - configuration, 5-11
- sidJCO2 job, 5-7
- sidJD00 job, 5-15
- sidPARM, 5-18
- SIGN SQL function, F-1
- SIMPLE linkage convention, 9-7
- SIMPLE WITH NULLS linkage convention, 9-7, G-1
- SIN SQL function, F-1
- SINH SQL function, F-1
- SITEINFO file, 6-8

- SLIP command, 10-9
- SMF
 - abends, D-3
 - abnormal terminations, D-3
 - activating, D-2
 - commands
 - SETSMF, D-3
 - default record types, D-3
 - interpreting records, D-3
 - ORAFMT program, D-4, D-8
 - record type, specifying, D-2
 - recording
 - starting, D-3
 - stopping, D-3
 - records, D-1
 - records, interpreting, D-3
 - sample formatting program, D-8
 - starting dynamically, D-3
 - types of users, D-2
 - user-defined data section, D-8
 - when records are written, D-3
- SMFAUTH authorization id, D-6, D-9
- SMFPRMxx member, D-2
- SMFWCPU field, 8-5
- snapshots, 9-17
 - complete refresh, 8-15
 - CREATE SNAPSHOT command, 8-15
 - Oracle snapshot feature, 8-8
- SNS/TCPaccess, 11-7
- software requirements, 3-2
 - IBM maintenance, 3-3
 - OS/390 operating system, 3-3
- SOUNDEX function, F-1
- specifying
 - SMF record type, D-2
- SPOOL
 - file, 5-37
 - space, 5-37
- SQL, 1-5
 - ANSI standard, 1-5
 - clauses
 - DELETE, 9-15
 - INSERT, 9-15
 - SELECT WHERE, 9-15
 - UPDATE, 9-15
 - commands, 8-2
 - constructs, Oracle-specific, 9-19
 - DB2, 9-7
 - DD script, 2-4
 - functions, 9-18, F-1
 - ABS, F-1
 - ACOS, F-1
 - ADD_MONTHS, F-1
 - advanced, 9-19
 - ASCII, F-1
 - ASIN, F-1
 - ATAN, F-1
 - ATAN2, F-1
 - AVG, 9-18
 - CEIL, F-1
 - CHAR_TO_ROWID, F-1
 - CHR, F-1
 - compatible, 9-18
 - compensated, 9-19
 - CONCAT, 9-18
 - CONVERT, F-1
 - COS, F-1
 - COSH, F-1
 - COUNT(*), 9-18
 - COUNT(DISTINCT expression), 9-18
 - DECODE, F-1
 - DUMP, F-1
 - EXP, F-1
 - FLOOR, F-1
 - GREATEST, F-2
 - HEXTORAW, F-2
 - 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
 - MAX, 9-18
 - MIN, 9-18

MOD, F-2
 MONTHS_BETWEEN, F-2
 NEW_TIME, F-2
 NEXT_DAY, F-2
 NLS_INITCAP, F-2
 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
 SINH, F-1
 SOUNDEX, F-1
 SQRT, F-1
 STDDEV, F-1
 SUBSTR, F-1
 SUBSTRB, F-2
 SUM, 9-18
 SYSDATE, F-2
 TAN, F-2
 TANH, 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, F-2
 translated, 9-18
 TRUNC, F-2
 UID, F-2
 UPPER, F-2
 USER, F-2
 USERENV, F-2
 VARIANCE, F-2
 VSIZE, F-2
 ISO standard, 1-5
 limitations, 2-5
 passthrough, 8-6, 9-7, 9-8, 9-9
 statements, 8-7, 9-2
 DB2, 9-8, 9-14
 passing through gateway, 9-7
 SQL*Plus, 1-6
 commands
 COPY, 8-9, 8-10
 DELETE, 9-20
 INSERT, 8-9
 SQLID, secondary id, 5-8
 SQLNET.EXPIRE_TIME parameter, 8-5
 SQLNET.ORA file, 8-5
 SQRT SQL function, F-1
 SRB (service request block)
 Oracle Net architecture, 6-3
 SRCLIB
 library, 7-9
 SRVRLMOD, 5-18
 START
 OSDI command
 defined, B-11
 starting Oracle Net service, 6-10
 START command, 7-2
 statements
 ALTER SESSION CLOSE DATABASE
 LINK, 8-3
 CONNECT TO, 8-3
 CREATE DATABASE LINK, 8-2
 DB2 CREATE TABLE, 1-5
 DDL, 9-8
 DROP DATABASE LINK, 8-3
 INSERT, 10-5
 ORA\$ENV DD, C-3
 SELECT, 8-6, 9-3, 9-19, 9-20
 SQL, 8-7, 9-2
 DB2, 9-8, 9-14
 passing through gateway, 9-7
 SYSMDUMP DD, 5-37
 STDDEV SQL function, F-1
 STEPLIB DD statement
 Oracle Net region JCL, 6-7
 STOP
 OSDI command, 6-11
 defined, B-14
 STOP (P) OS/390 System command, 6-11
 STOP command, 7-2

- storage requirements, 3-2
- stored procedures, 1-4, 1-5
 - AUTHID column, 9-5
 - creating on DB2, 9-4
 - DB2, 8-6, 9-4, 9-6
 - CURRENT SERVER value, 9-5
 - PROC1, 9-5
 - using, 9-4
 - DB2IND, G-1
 - error message, 10-4
 - executing, 9-5
 - LUNAME column, 9-5
 - native DB2, 1-6
 - Oracle, 1-6
 - Oracle database server
 - local instance, 9-3
 - PL/SQL, 9-3
 - remote instance, 9-3
 - using, 9-3
 - ORAIND, G-3
 - parameter list, 9-6
 - PROCEDURE column, 9-5
 - SYSPROC qualifier, 9-6
 - SYS PROCEDURES catalog table, 9-5
 - two-phase commit processing, 9-4
- stored procedures in DB2, gateway features, 2-2
- Streams Heterogeneous Replication, 2-3
- SUBSTR SQL function, F-1
- SUBSTRB SQL function, F-2
- SUM
 - option, 9-20
 - Oracle database server function, 9-20
 - SQL function, 9-18
- support
 - sending tapes, 10-5
- synonym feature, 8-6
- SYS1.PARMLIB
 - library, 4-22, 4-23
- SYSABEND for MPM startup, 5-37
- SYS DATE SQL function, F-2
- SYSIBM.SYS PROCEDURES DB2 catalog table, 9-5
- SYSMDUMP DD statement, 5-37
- SYSOUT
 - data set, 5-37
- SYS PROCEDURES catalog table, 9-5, 9-6

- system
 - dump data sets, 10-10
 - requirements
 - resource, 3-2
 - software, 3-2
- SYSUDUMP for MPM startup JCL, 5-37

T

- TABLE_PRIVILEGES view, E-10
- tables
 - ASCII, 2-4
 - DB2, 2-4, 8-16, 9-11, 9-12
 - EBCDIC, 2-4
 - ORACLE2PC, 2-4, 8-8
 - SYSIBM.SYS PROCEDURES DB2 data
 - dictionary, 9-7
 - SYS PROCEDURES, 9-5
- TAN SQL functions, F-2
- TANH SQL function, F-2
- tape
 - contents, 3-5
- TARGET (sidENV variable), 5-25
- TCB, 7-9
- TCP/IP
 - KEEPALIVE, 8-5
- TESTAUTH link, 5-7
- threads
 - canceling, 8-5
 - managing, 8-4
- TIME data type, 9-12
- time operations, 9-12
- TIMESTAMP data type, 9-12
- TNS
 - connect descriptor, 8-2
- TNS facility, 11-2
- TNS messages, obsolescence of, 11-3
- TNS NAMES SID
 - parameter, 2-3
- TNS NAMES.ORA
 - connect descriptor, 8-2
- TO_CHAR
 - SQL function, F-2
- TO_DATE
 - clause, 9-16

- function, 9-14, 9-15, 9-16
- option, 9-20
- Oracle database server function, 9-20
- SQL function, F-2
- TO_LABEL function, F-2
- TO_MULTI_BYTE SQL function, F-2
- TO_NUMBER SQL function, F-2
- TO_NUMBER_OFF (sidENV variable), 5-27
- TO_SINGLE_BYTE SQL function, F-2
- Trace Files
 - OSDI, 11-3
- TRACELEVEL (sidENV variable), 5-27
- TRANSLATE SQL function, F-2
- translated SQL function, 9-18
- TRCASST, Oracle Net utility program, 6-12
- triggers, 8-8
 - database, 1-4
 - DB2 server, 8-9
 - Oracle database server, 8-9
- TRUNC SQL function, F-2
- two-phase commit, 1-4, 8-8
- TYPE (gateway service DEFINE parameter), 5-30

U

- UID SQL function, F-2
- UNION capability, 1-3
- UNIX environment, 8-4
- UPDATE SQL clause, 9-15
- UPPER SQL function, F-2
- USER
 - SQL function, F-2
- user
 - exit module, 7-6
 - record type, D-2
- USER_CATALOG view, E-10
- USER_COL_COMMENTS view, E-11
- USER_CONS_COLUMNS view, E-11
- USER_CONSTRAINTS view, E-11
- USER_DB_LINKS data dictionary view, 8-3
- USER_INDEXES view, E-12
- USER_OBJECTS view, E-13
- USER_SYNONYMS view, E-13
- USER_TAB_COLUMNS view, E-14
- USER_TAB_COMMENTS view, E-14

- USER_VIEWS view, E-16
- USERENV function, F-2
- userid
 - CONNECT TO, 8-3
 - OTGDB2, 5-6
- using
 - the data dictionary, 9-22
- USING clause, 8-2
- USS (OS/390 UNIX System Services)
 - with network service, 6-8
- utilities
 - IEBUPDTE, 4-22

V

- VALUES clause, 9-16
- VARIANCE SQL function, F-2
- views
 - catalog for DB2, 9-22
 - data dictionary, 9-22
 - ALL_CATALOG, E-2
 - ALL_COL_COMMENTS, E-2
 - ALL_CON_COLUMNS, E-3
 - ALL_CONSTRAINTS, E-3
 - ALL_DB_LINKS, 8-3
 - ALL_IND_COLUMNS, E-4
 - ALL_INDEXES, E-4
 - ALL_OBJECTS, E-5
 - ALL_SYNONYMS, E-6
 - ALL_TAB_COLUMNS, E-6
 - ALL_TAB_COMMENTS, E-7
 - ALL_TABLES, E-7
 - ALL_USERS, E-8
 - ALL_VIEWS, E-9
 - COLUMN_PRIVILEGES, E-9
 - DBA_DB_LINKS, 8-3
 - OTGREGISTER, 9-22, E-9
 - TABLE_PRIVILEGES, E-10
 - USER_CATALOG, E-10
 - USER_COL_COMMENTS, E-11
 - USER_CONS_COLUMNS, E-11
 - USER_CONSTRAINTS, E-11
 - USER_DB_LINKS, 8-3
 - USER_INDEXES, E-12
 - USER_OBJECTS, E-13

- USER_SYNONYMS, E-13
- USER_TAB_COLUMNS, E-14
- USER_TAB_COMMENTS, E-14
- USER_VIEWS, E-16
- virtual memory requirements, 3-2
- volume serial number specification, 4-6
- volume unit specification, 4-7
- VSAM
 - file, 9-7
- VSIZE SQL function, F-2

W

- WE8EBCDIC37C with NLS, C-4
- WHERE clause, 8-15, 9-19
- WHERE CURRENT OF CURSOR clause, 9-17
- wireless communication, 1-4
- Workload Manager (WLM)
 - Oracle Net for OS/390 architecture, 6-3