Oracle® Transparent Gateway for DB2/400

Installation and User’s Guide

Release 9.2.0.1.0 for IBM AS/400

July 2002
Part No. A97615-01
# Contents

Send Us Your Comments ................................................................................................................... xi

Preface ........................................................................................................................................................ xiii

1 Introduction

| Oracle Transparent Gateway for DB2/400 Introduction | 1-1 |
| Protection of Current Investment | 1-1 |
| **Release 9i Gateways** | 1-3 |
| Advantages of the Gateway | 1-3 |
| SQL | 1-7 |
| Stored Procedures | 1-7 |
| Languages | 1-8 |
| SQL*Plus | 1-8 |
| Oracle9i Database Server Technology and Tools | 1-8 |
| **Two-Phase Commit and Multi-site Transactions** | 1-8 |
| Site Autonomy | 1-9 |
| Migration and Coexistence | 1-9 |
| Security | 1-9 |
| Gateway Architecture | 1-9 |
| **How the Gateway Works** | 1-10 |
| Gateway Changes to Your AS/400 System | 1-11 |
| SQL Differences | 1-13 |
| Heterogeneous Services Architecture | 1-13 |
2 Release Information

Product Set ............................................................................................................... 2-2
Changes and Enhancements .................................................................................... 2-2
Known Problems ...................................................................................................... 2-4
Known Restrictions .................................................................................................. 2-5
Owners of DB2/400 Components ............................................................................ 2-7
DB2/400 SQL Limitations ....................................................................................... 2-8

3 System Requirements

Hardware Requirements ............................................................................................ 3-2
  Processor .................................................................................................................. 3-2
  Memory ....................................................................................................................... 3-2
  CD-ROM Drive .......................................................................................................... 3-2
  Disk Space .................................................................................................................. 3-3
Software Requirements ............................................................................................ 3-3
  Operating System Requirements ............................................................................. 3-3
  Oracle Integrating Server ........................................................................................ 3-3
Documentation Requirements ................................................................................... 3-3
  Distribution Kit ......................................................................................................... 3-3

4 Installing the Gateway

Gateway Co-existence ............................................................................................... 4-2
Checklists .................................................................................................................... 4-3
  Pre-Installation Checklist ....................................................................................... 4-3
  Installation Checklist ............................................................................................... 4-3
  Post-Installation Checklist ..................................................................................... 4-3
Preinstallation Steps .................................................................................................. 4-4
  Step 1: Log on ............................................................................................................. 4-4
  Step 2: System Value for the QCRTAUT Parameter ............................................... 4-4
Installation Steps

Step 1: Mount the CD-ROM ................................................................. 4-6
Step 2: Start the Install Process ........................................................ 4-6
Step 3: Set Required Parameters ....................................................... 4-6
Step 4: Verify Installation ................................................................. 4-9
Step 5: Finish the Installation ............................................................ 4-10
Step 6: Start the Gateway Subsystem ............................................... 4-10

Post-Installation Steps

Journalling ......................................................................................... 4-11
Step 1: Verify the Gateway Version .................................................. 4-12
Step 2: Verify Journalling ................................................................. 4-12
Step 3: Journal the ORACLE2PC File .............................................. 4-13
Step 4: Configure Oracle Net ............................................................ 4-14
Step 5: Remove Observability ........................................................... 4-14
Step 6: Raise the Gateway Priority Level ........................................ 4-14

5 Oracle Net

Overview of Oracle Net........................................................................ 5-2
Distributed Processing ........................................................................ 5-2
Distributed Database ........................................................................... 5-2
Terminology for Oracle Net ............................................................... 5-3
Oracle Net for AS/400 Architecture .................................................. 5-3
Oracle Net File Name Structure ....................................................... 5-5

Checklists for Configuring Oracle Net .............................................. 5-7
Checklist for Configuring Oracle Net for TCP/IP .............................. 5-7
Connecting through TCP/IP (on the Oracle Server) Checklist ................ 5-7
Preliminary Step: Define a Physical Line ......................................... 5-8

Configuring Oracle Net for TCP/IP AS/400 ...................................... 5-9
Configuring for TCP/IP .................................................................... 5-10
Step 1: Modify the Port Number ...................................................... 5-10
Step 2: Define the Host Name ......................................................... 5-11
Step 3: Verify the Host Name ......................................................... 5-12
Step 4: Verify that the Listener is Started ....................................... 5-12
Connecting to the AS/400 through TCP/IP ................................................................. 5-12
  Step 1: Add a TCP/IP Connect Descriptor to TNSNAMES.ORA ............................. 5-13
  Step 2: Specify SID Names....................................................................................... 5-14
Troubleshooting TCP/IP Connection Problems ......................................................... 5-15
  Resolving Connection Problems............................................................................. 5-15
  Troubleshooting Your TCP/IP Configuration.......................................................... 5-17

6  Configuring the Gateway

Gateway Commands .................................................................................................... 6-2
  Running the Commands ............................................................................................. 6-3
  CRTORAGTWI, Copy the Gateway ......................................................................... 6-4
  CHGORANET, Change Network Parameters ......................................................... 6-8
  CHGORAPJE, Change Prestarted Job Parameters ................................................. 6-10
  CHGORATUN, Change Initialization Parameters .................................................. 6-10
  CHGGTWDBG, Change Debugging Parameters .................................................... 6-14
  CRTORADD, Create Data Dictionary Views ........................................................... 6-15
  CHGRECOPRF, Change Recovery Profile Parameters ........................................... 6-16

Setting Optional Parameters ....................................................................................... 6-18
  Retrieving Data ........................................................................................................ 6-18
  Data Conversion ....................................................................................................... 6-19
  Read-Only Gateway .................................................................................................. 6-19

7  Using the Gateway

Database Link Behavior ............................................................................................... 7-2
  Creating Database Links ......................................................................................... 7-2
  Creating Database Links Using Oracle Net ............................................................ 7-3
  Closing Database Links ......................................................................................... 7-3
  Accessing Data through Database Links .............................................................. 7-4
  Dropping Database Links ..................................................................................... 7-4
  Examining Available Database Links ................................................................... 7-4
  Limiting the Number of Active Database Links .................................................... 7-5

Using DB2/400 Cursors ............................................................................................. 7-5
Using the Synonym Feature ...................................................................................... 7-5
Accessing the Gateway .............................................................................................. 7-6
  Viewing Jobs ............................................................................................................. 7-7
# Accessing AS/400 File Members

- Accessing Flat Files ................................................................. 7-7

# Performing Distributed Queries

- Example of a Distributed Query ............................................. 7-8
- Two-Phase Commit Processing .............................................. 7-9

# Replicating in a Heterogeneous Environment

- Oracle9i database server Triggers ........................................ 7-11
- Oracle Snapshots ................................................................. 7-11

# Copying Data from the Oracle9i database server to the DB2/400 Server

- Triggers .................................................................................. 7-12
- SQL*Plus COPY Command ..................................................... 7-13

# Copying Data to the Oracle9i database server from the DB2/400 Server

- Triggers .................................................................................. 7-12
- SQL*Plus COPY Command ..................................................... 7-13

## Administering the Gateway

- Overview of the Oracle Gateway Monitor .............................. 8-2
- Starting the Oracle Gateway Monitor .................................... 8-2

## Developing Applications

- Gateway Appearance to Application Programs ...................... 9-2
  - Connecting to DB2/400 Databases with Database Links ........ 9-2
  - Read and Write Access to DB2/400 Databases ..................... 9-3
  - Accessing Tables in Multiple Databases ............................ 9-3

## Array Processing

- Fetch Reblocking ................................................................. 9-4

## Using Oracle Stored Procedures with the Gateway

- ................................................................. 9-5
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using DB2/400 Stored Procedures with the Gateway</td>
<td>9-7</td>
</tr>
<tr>
<td>Executing DB2/400 Stored Procedures from Applications</td>
<td>9-8</td>
</tr>
<tr>
<td>Considerations for Using Stored Procedures with DB2/400</td>
<td>9-10</td>
</tr>
<tr>
<td>Passing DB2/400 SQL Statements through the Gateway</td>
<td>9-11</td>
</tr>
<tr>
<td>Using the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE Function</td>
<td>9-11</td>
</tr>
<tr>
<td>Examples</td>
<td>9-12</td>
</tr>
<tr>
<td>Using Passthrough Statements to Create Tables</td>
<td>9-13</td>
</tr>
<tr>
<td>Retrieving Results Sets Through Passthrough</td>
<td>9-14</td>
</tr>
<tr>
<td>Converting DB2/400 Datatypes to Oracle Datatypes</td>
<td>9-15</td>
</tr>
<tr>
<td>DB2/400 GRAPHIC Support</td>
<td>9-17</td>
</tr>
<tr>
<td>For DBCS Users</td>
<td>9-17</td>
</tr>
<tr>
<td>Performing Character String Operations</td>
<td>9-19</td>
</tr>
<tr>
<td>Converting Character String Datatypes</td>
<td>9-20</td>
</tr>
<tr>
<td>Performing Date and Time Operations</td>
<td>9-20</td>
</tr>
<tr>
<td>Dates in the 21st Century</td>
<td>9-22</td>
</tr>
<tr>
<td>Performing Numeric Datatype Operations</td>
<td>9-22</td>
</tr>
<tr>
<td>SQL Functions</td>
<td>9-23</td>
</tr>
<tr>
<td>Oracle Server SQL Construct Processing</td>
<td>9-25</td>
</tr>
<tr>
<td>SELECT without the FOR UPDATE Clause</td>
<td>9-25</td>
</tr>
<tr>
<td>SELECT FOR UPDATE, INSERT, and DELETE Clauses</td>
<td>9-26</td>
</tr>
<tr>
<td>Oracle Server and DB2/400 Differences</td>
<td>9-27</td>
</tr>
<tr>
<td>Mass Delete from a Segmented Tablespace</td>
<td>9-27</td>
</tr>
<tr>
<td>Oracle Bind Variables</td>
<td>9-27</td>
</tr>
<tr>
<td>Mapping the COUNT Function</td>
<td>9-27</td>
</tr>
<tr>
<td>Performing Zoned Decimal Operations</td>
<td>9-28</td>
</tr>
<tr>
<td>Oracle Data Dictionary Emulation in a DB2/400 Server</td>
<td>9-28</td>
</tr>
<tr>
<td>Using the Gateway Data Dictionary</td>
<td>9-28</td>
</tr>
<tr>
<td>DB2/400 Special Registers</td>
<td>9-28</td>
</tr>
<tr>
<td>Oracle Developer Forms Compatibility</td>
<td>9-29</td>
</tr>
</tbody>
</table>

10 Error Messages, Diagnosis, and Reporting

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message and Error Code Processing</td>
<td>10-2</td>
</tr>
<tr>
<td>Mapping DB2/400 Error Messages to Oracle Error Messages</td>
<td>10-2</td>
</tr>
<tr>
<td>Interpreting Gateway Message Formats</td>
<td>10-2</td>
</tr>
<tr>
<td>Diagnosing Errors Detected by the Oracle9i database server</td>
<td>10-4</td>
</tr>
</tbody>
</table>
### Contacting Oracle Support Services

- Error Categories
  - Documentation Errors
  - Incorrect Output
  - Oracle External Error
  - Performance
  - Missing Functionality

### Oracle Net Files and Members

- LOG File Members
- ORA File Members
- TRC File Members
- Agent Trace Files

### National Language Support

- Overview of National Language Support
- Gateway Configuration
  - CHGORATUN Language Specification
- DB2/400 Coded Character Set Considerations
  - Supported Character Sets
  - Supported Languages and Territories
- Oracle Server and Client Configuration
- Message Availability
- DB2/400 GRAPHIC Support

### Data Dictionary Views

- ALL_CATALOG
- ALL_COL_COMMENTS
- ALL_CONSTRAINTS
- ALL_IND_COLUMNS
- ALL_INDEXES
- ALL_OBJECTS
- ALL_SYNONYMS
- ALL_TAB_COLUMNS
ALL_TAB_COMMENTS ................................................................................................................ C-9
ALL_TABLES ............................................................................................................................ C-9
ALL_USERS .............................................................................................................................. C-10
ALL_VIEWS ............................................................................................................................ C-11
COLUMN_PRIVILEGES ........................................................................................................ C-11
OTGREGISTER ........................................................................................................................ C-12
TABLE_PRIVILEGES ................................................................................................................ C-13
USER_CATALOG ...................................................................................................................... C-13
USER_COL_COMMENTS ......................................................................................................... C-13
USER_CONS_COLUMNS ........................................................................................................ C-14
USER_CONSTRAINTS ............................................................................................................... C-14
USER_INDEXES ....................................................................................................................... C-15
USER_OBJECTS ........................................................................................................................ C-16
USER_SYNONYMS .................................................................................................................. C-17
USER_TAB_COLUMNS ............................................................................................................. C-18
USER_TAB_COMMENTS ......................................................................................................... C-19
USER_TABLES ........................................................................................................................ C-19
USER_USERS .......................................................................................................................... C-21
USER_VIEWS .......................................................................................................................... C-21
Send Us Your Comments

Oracle Transparent Gateway for DB2/400 Installation and User's Guide, Release 9.2.0.1.0 for IBM AS/400
Part No. A97615-01

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

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

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

   infoibm_us@oracle.com

Include your name, address, and telephone number.

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

Read this guide if you are responsible for performing tasks such as:
- installing and configuring the gateway
- administering the gateway
- using the gateway

Understand the fundamentals of transparent gateways and AS/400 before using this guide for installation or system administration.

Intended Audience

This guide is intended for anyone installing, configuring and administering the gateway, as well as for developers writing applications that access remote host databases through the gateway.
Documentation Accessibility

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

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

Product Name

The complete name for this product is Oracle Transparent Gateway for DB2/400, also abbreviated TG4DB2/400 in this document.
Conventions

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

\texttt{WRKCFGSTS *DEV device\_name}

Punctuation, such as commas and quotes, must be entered as shown unless otherwise specified. All output is shown as it actually appears. For input, these conventions apply:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPPERCASE</td>
<td>Uppercase words, such as \texttt{WRKCFGSTS}, indicate a word or phrase must be entered exactly as spelled.</td>
</tr>
<tr>
<td>lowercase code font</td>
<td>Lowercase, such as \texttt{device_name}, indicates a word or phrase must be substituted, such as the actual device name.</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Angle brackets indicate the enclosed arguments are required, and at least one of the arguments must be entered. Do not enter the brackets themselves.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Square brackets indicate the enclosed arguments are optional. Do not enter the brackets themselves</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>Ellipses indicate the preceding item can be repeated. You can enter an arbitrary number of similar items.</td>
</tr>
</tbody>
</table>

Commands, reserved words, and keywords appear in uppercase in both examples and text. A file ID can appear with both upper- and lowercase text. The portions of a file ID appearing in lowercase indicate those portions can vary. Reserved words and keywords must always be entered as is and have reserved meanings within Oracle.

Storage Measurements

Storage measurements use these abbreviations:

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

**Oracle Books:**
- Oracle9i Administrator’s Reference
- Oracle Advanced Security Administrator’s Guide
- Oracle9i Application Developer’s Guide - Fundamentals
- Oracle C++ Call Interface Programmer’s Guide
- Oracle Call Interface Programmer’s Guide
- Oracle9i Database Administrator’s Guide
- Oracle9i Database Error Messages
- Oracle9i Database Reference
- Oracle9i Database Utilities
- Oracle9i Heterogeneous Connectivity Administrator’s Guide
- Oracle9i Net Services Administrator’s Guide
- Oracle9i Net Services Reference Guide
- Oracle9i SQL Reference
- Oracle Universal Installer Concepts Guide
- PL/SQL User’s Guide and Reference
- SQL*Plus User’s Guide and Reference

**IBM Books:**
Refer to the IBM documents for your platform and operating system.
Oracle Transparent Gateway for DB2/400 Introduction

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

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

Protection of Current Investment

Oracle Transparent Gateway for DB2/400 gives your company the ability to develop its information systems without forfeiting your investments in current data and applications. The gateway gives you access to your Oracle and DB2/400 data with a single set of applications while you continue to use existing IBM applications to access your IBM data. You can also use more productive database tools and move to a distributed database technology without giving up access to your current data.

If you choose to migrate to Oracle database technology and productivity, the gateway allows you to control the pace of your migration. As you transfer applications from your previous technology to the Oracle database, you can use the gateway to move the DB2/400 data into Oracle databases.
The Oracle Transparent Gateway for DB2/400 enables you to:
- Integrate heterogeneous database management systems so that they appear as a single homogeneous database system.
- Read and write data from Oracle applications to data in DB2/400, in addition to any Oracle server data.

This chapter presents information about the architecture, uses, and features of the Oracle Transparent Gateway for DB2/400. It contains the following sections:

- **Release 9i Gateways** on page 1-3
- **Two-Phase Commit and Multi-site Transactions** on page 1-8
- **Site Autonomy** on page 1-9
- **Migration and Coexistence** on page 1-9
- **Security** on page 1-9
- **Gateway Architecture** on page 1-9
- **How the Gateway Works** on page 1-10
Release 9i Gateways

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

Release 9i gateways are tightly integrated with the Oracle9i database server, enabling improved performance and enhanced functionality while still providing transparent integration of Oracle data and non-Oracle data. For example, connection initialization information is available in the local Oracle9i database server, reducing the number of round trips and the amount of data that is sent over the network. Structured Query Language (SQL) execution is also faster, because statements issued by an application are parsed and translated once and can then be reused by multiple applications.

Release 9i 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 DB2/400 enables Oracle client applications to access DB2/400 through SQL. The gateway and Oracle9i database server together create the appearance that all data resides on a local Oracle9i database server, even though data might be widely distributed. If data is moved from a DB2/400 database to an Oracle database, then no changes in client application design or function are needed. The gateway handles all differences in datatypes and SQL functions between the application and the database.

Oracle Transparent Gateway for DB2/400 gives you the power to integrate your heterogeneous systems 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 world-class tools, networking, and data server technology, the Oracle Transparent Gateway for DB2/400 sets the standard for seamless, enterprise-wide information access.
Transparency at All Levels

By using the Oracle Transparent Gateway for DB2/400, 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 the Oracle Net technology to allow users to access data across multiple networks, without concern for the network architecture or protocols. Multiple protocols are supported.

- Operating system transparency:
  You can access data that is stored under multiple operating systems without being aware of the different operating systems that hold the data.

- Data storage transparency:
  Data can be accessed regardless of the database or file format.

- Access method transparency:
  You can utilize a single dialect of SQL for any data store, eliminating the need to code for database-specific access methods or SQL implementations.

Extended Database Services

The 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, which 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 that the newest features of each database release are always available immediately to the gateway.

- Distributed capabilities:
  Heterogeneous data can be integrated seamlessly because Oracle database 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 that SQL statements are executed 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 that a transaction that spans data stores is still treated as a single unit of work. Changes are not committed, nor 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 of your data, ensuring uniform enforcement of your business rules across the enterprise.

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

Implicit protocol conversion:
Oracle and Oracle Net can work together as a protocol converter, allowing applications to transparently access other data stores on platforms that do not support the client network protocol. For example, an application can use SPX/IPX to communicate with an Oracle9i database server, which can use TCP/IP to communicate with the gateway and another data store on a non-SPX/IPX platform.
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 Security.

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

Dynamic Dictionary Mapping
Before an application can access any information, the application must be told the structure of the data, such as the names of columns of a table and their lengths. Many other products require administrators to manually define this information in a separate data dictionary stored in a hub. Applications then 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 on your part. As administrators, you must update the data dictionary in the hub whenever the structure of a remote table is changed.

Inefficient duplication is eliminated with Oracle Transparent Gateway for DB2/400. The simple setup of the gateway does not require any additional mapping. 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 need ever be created or maintained.
SQL

Oracle Transparent Gateway for DB2/400 eases 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.

Passthrough and Native DB2/400 SQL

Native DB2/400 SQL can be passed through the gateway for running directly against DB2/400. This enables applications to send statements, such as a DB2/400 CREATE TABLE, to the gateway for execution on a target DB2/400 system.

Data Definition Language

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

Data Control Language

You can issue native data control language (DCL) statements from an Oracle environment, allowing central administration of user privileges and access levels for heterogeneous data stores. Refer to Oracle9i Heterogeneous Connectivity Administrator’s Guide for more information on DCL.

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, with no special coding for the heterogeneous data access.

Oracle Stored Procedures

Oracle stored procedures enable you to access and update DB2/400 data using centralized business rules that are stored in the Oracle9i database server. The use of 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 DB2/400 Stored Procedures

The gateway can execute DB2/400 stored procedures using standard Oracle PL/SQL. The Oracle application executes the DB2/400 stored procedure as if it were an Oracle remote procedure.

**Note:** If you are going to use DB2/400 stored procedures, then you must journal the ORACLE2PC table.

Languages

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 ad-hoc query tools, web browsers, turnkey applications, and application development tools.

SQL*Plus

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

Oracle9i Database Server Technology and Tools

The gateway is integrated into the Oracle9i database server technology, which provides global query optimization, transaction coordination for multi-site transactions, support for all Oracle Net configurations, and so on. Tools and applications that support the Oracle9i database server can be used to access heterogeneous data through the gateway.

Two-Phase Commit and Multi-site Transactions

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

- a full two-phase commit partner
- a commit point site
Gateway Architecture

Introduction

- a single-site update partner
- a read-only partner

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

Oracle Transparent Gateway for DB2/400, by default, is configured as a commit point site (that is, commit-confirm protocol). Optionally, you can configure the gateway as read-only if you choose to enforce read-only capability through the gateway. Other protocols are not supported. Refer to "Read-Only Gateway" on page 6-19 for more information.

Site Autonomy

All Oracle9i database server products, including gateways, supply site autonomy. For example, administration of a data source remains the responsibility of the original system administrator. Site autonomy also functions such that gateway products do not override the security methods of the data source or the operating environment.

Migration and Coexistence

The integration of a data source through the gateway requires no changes to be made to applications at the data source. As a result of this, the Oracle9i 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 that are already used in the operating environment of the data source.

Functionally, gateway security is identical to that of an Oracle9i database server, as described in the Oracle9i Administrator’s Reference. Oracle database security is mapped to the data dictionary of the data source.

Gateway Architecture

The gateway architecture consists of four main components as shown in Figure 1–1.
How the Gateway Works

1. Client
   The client is an Oracle application or tool.

2. Oracle9i database server:
   The Oracle9i database server is an Oracle instance.

3. Oracle Transparent Gateway for DB2/400:
   The Oracle Transparent Gateway for DB2/400 must be installed on an AS/400 system, and is responsible for issuing the dynamic SQL calls to DB2/400. The gateway can access files in an OS/400 collection or files that are externally described to DB2/400.

   The gateway is not started as an Oracle instance and has no continuously running background processes. The gateway is started either using the OS/400 STRSBS command or by using the ORAMON command. Individual gateway tasks run in a subsystem, as does a LISTENER task if TCP/IP is used.

4. DB2/400 Server:
   The DB2/400 database is the database that is being accessed by the gateway.

   Multiple Oracle9i database servers can access the same gateway. A single gateway installation can be configured to access one, and only one, DB2/400 server, because there is only one instance of a DB2/400 database per AS/400 system.

Figure 1–1 Architecture of the Oracle Transparent Gateway for DB2/400

How the Gateway Works

The gateway has no database functions of its own. Instead, it provides an interface by which the Oracle9i database server can direct SQL operations to a DB2/400 database.
The gateway that is supporting the DB2/400 server is identified to the Oracle9i database server using a database link. The database link is the same construct used to identify other Oracle9i database server databases.

Tables on the DB2/400 server are referenced in SQL as:

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

or, preferably, as:

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

If you create synonyms or views in the Oracle database, you can refer to tables on the DB2/400 server by using simple names as though the tables were local to the Oracle9i database server.

When the Oracle9i database server encounters a reference to a table on the DB2/400 server, the applicable portion of the SQL statement is sent to the gateway for processing. Any host variables that are associated with the SQL statement are bound to the gateway and, therefore, to the DB2/400 server.

The gateway is responsible for sending these SQL statements to the DB2/400 server. The DB2/400 server is responsible for executing the SQL statements and for fielding and returning responses.

**Gateway Changes to Your AS/400 System**

Installing the gateway on your AS/400 system produces the following changes to the system:

1. A library is created, with the instance name as given in the first install panel. (Refer to Figure 4–1 on page 4-7 for an example). Almost all of the gateway will be found in this library, with the exception of some objects dealing with service programs and the items mentioned in parts 3 and 4 of this list.

   The gateway can also be installed into a previously-created DB2/400 SQL COLLECTION that contains nothing but logical files.

2. A user profile is created. The user profile has the same name as the library that is created when you install the gateway.

3. A subdirectory entry is created in the /home directory in the Integrated File System. The subdirectory entry name will be the same as the instance name. Thus, if ORACLE is the instance name, then a subdirectory node will be created at /home/ORACLE. Further nodes will be created below this node. Agent trace
files will appear (if requested) in a subdirectory of the created node. An initialization file will also appear in a subdirectory of this node.

4. A library named ORASRVLIB is created. If the install software finds that it must install the service programs that are found on the install medium, then a library with a name such as ORASRVnnnn (where nnnn is a number) will be created. The relevant service programs and a file that is used for NLS transactions will be placed in that library. Additional files or members within files may be added to the ORASRVLIB library.

5. A message queue with the name of the install library is created in the QUSRSYS library. This is as a result of using the CRTUSRPRF command to create the user profile that corresponds to the install library.
SQL Differences

Not all SQL implementations are the same. The Oracle9i database server may support a larger set of built-in functions than the databases that are currently accessed through the gateway. Or, the semantics of some functions in the DB2/400 server may not agree with an identically named function in the Oracle9i database server. The SUBSTR function is an example of this, since the second and third arguments of SUBSTR in an Oracle9i database server may be negative. In DB2/400, the second and third arguments of the SUBSTR function cannot be negative. In any case, the Oracle9i database server and the gateway work together to convert the SQL to a form that is compatible with the specific DB2/400 server.

During this conversion, an Oracle9i database server function can be converted to a function recognizable to the specific DB2/400 server. For example, the Oracle9i database server NVL function is converted to the DB2/400 VALUE function.

Alternatively, the Oracle integrating server withholds functions that are not executable by the DB2/400 server and performs the functions on the Oracle side after the rows are fetched from the DB2/400 server. This is called post-processing because it occurs after the rows are fetched from DB2/400. This processing generally applies to SELECT statements. The Oracle integrating server and the gateway cannot perform this kind of manipulation on UPDATE, INSERT, or DELETE statements because doing so changes transaction semantics. The SUBSTR function is an example of a function that is always post-processed on the Oracle server side.

Heterogeneous Services Architecture

This release of the Oracle Transparent Gateway for DB2/400 utilizes the Oracle Heterogeneous Services component within the Oracle9i server. Heterogeneous Services is the building block for the next generation of Oracle Open Gateways. For detailed information about heterogeneous services, refer to the Oracle9i Heterogeneous Connectivity Administrator’s Guide.
How the Gateway Works
This chapter describes the changes and corrected problems in this release. It contains the following sections:

- **Product Set** on page 2-2
- **Changes and Enhancements** on page 2-2
- **Known Problems** on page 2-4
- **Known Restrictions** on page 2-5
- **Owners of DB2/400 Components** on page 2-7
- **DB2/400 SQL Limitations** on page 2-8
Product Set

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

- Oracle Transparent Gateway for DB2/400, release 9.2.0.1.0
- Oracle Net, release 9.2.0.1.0

Changes and Enhancements

SQL*Plus DESCRIBE Command

The SQL*Plus version 9.2 DESCRIBE command can be used to access remote DB2/400 object information.

Support for DBCS

This support applies to Eastern Asian Languages.

Support for UCS-2 (Unicode)

Unicode columns in DB2/400 tables can now be processed by the gateway.

Enhancements to Support the Turkish Character Set (CCSID 1026)

CCSID 1026 is now defined within DB2/400.

Recovery Password Encryption

The recovery password is no longer in plain text. For more information, refer to "CHGRECPRF, Change Recovery Profile Parameters" on page 6-16.

Table 2–1  Bugs Fixed Since Release 8.1.7.0 of the Gateway

<table>
<thead>
<tr>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2387621:</td>
<td>SUPPORT FOR CHARACTER SET UCS-2 REQUIRED IN TG4DB2/400</td>
</tr>
<tr>
<td></td>
<td>Product now supports reference of columns declare with CCSID 13488, which is IBM's UCS-2 level 1.</td>
</tr>
<tr>
<td>2314194:</td>
<td>LOGTRCFILJ NOT DUPLICATED BY CRTORAGTWI</td>
</tr>
<tr>
<td></td>
<td>The LOGTRCFILJ *FILE object is now duplicated when an instance is cloned.</td>
</tr>
<tr>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 2275120    | ORA-28527: HETEROGENEOUS SERVICES DATATYPE MAPPING ERROR WHEN QUERYING ALL_VIEWS  
New definition of the ALL_VIEWS Data Dictionary item |
| 2158882    | NULL SQLPLUS OUTPUT USING NVL OPERATOR AGAINST A TG4DB2/400 VIEW/TABLE  
LTRIM and RTRIM are now post-processed |
| 2125172    | AS/400 BINARY FIELD INCORRECTLY CONVERTED WITH VERSION 8 OF THE GATEWAY  
Precision and Scale are now taken into account for Binary fields |
| 2117335    | ORA-2068 AND ORA-28511 WHEN SELECTING A NUMBER COLUMN WITH SPACE INIT  
MCH1202/CEE9901 on attempted of conversion of invalid data in ZONED columns (invalid data came from legacy file) |
| 2113830    | ZONED DECIMAL DATA NOT DISPLAYED CORRECTLY USING DSPPFM  
Gateway not using "normal" sign nibble value in ZONED data. |
| 2043393    | Security fixes in Oracle Net |
| 2043386    | Security fixes in Oracle Net |
| 2030307    | SIGNON BY THE GATEWAY USER FAILED AT ORAPROFILE 1500  
This is a DBCs problem. Occurred at customer running under CCSID 5026. |
| 2018853    | CT HAS TWO ETHERNET CARDS; WANTS LISTENER TO LISTEN ON BOTH  
This really is not a bug but more of a non-normal configuration problem that has come up many times in the past. You need to manually update the ORA(LISTENER) member in the instance (even though we say you really should not be editing that member directly). Following is a sample for a two-port listener: |

```sql
LISTENER=
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=TCP)(HOST=138.67.159.198)(PORT=03817))
   (ADDRESS=(PROTOCOL=TCP)(HOST=136.192.120.81)(PORT=13817))
  )
```
Known Problems

The problems documented in the following section are specific to the Oracle Transparent Gateway for DB2/400, and are known to exist in this release of the product. These problems are currently being addressed by Oracle Corporation. Refer to the respective bugs for the current status of each problem. If you have any questions or concerns about these problems, please contact Oracle Support Services.

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

---

**Table 2–1 (Cont.) Bugs Fixed Since Release 8.1.7.0 of the Gateway**

<table>
<thead>
<tr>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000560:</td>
<td>MESSAGE APPEARED ON INSTALLING TG4DB2400 R8.1.7</td>
</tr>
<tr>
<td></td>
<td>This is a DBCS problem. Message was 'Could not open file &quot;/home/T8170T/rdbms/admin/initOTG_INSTANCE.ora&quot; To finish the terminal session, push execute key.'</td>
</tr>
<tr>
<td></td>
<td>Customer running under CCSID 5026</td>
</tr>
<tr>
<td>1897096:</td>
<td>CPF2105/CPF2130 ON CRTDUPOBJ FOR QPRINT *FILE OBJECT</td>
</tr>
<tr>
<td></td>
<td>Error when cloning an instance</td>
</tr>
<tr>
<td>1897075:</td>
<td>CPF1023 FROM CRTORAGTWN WHEN CLONING AN INSTANCE</td>
</tr>
<tr>
<td>1897030:</td>
<td>CPF1023 FROM CRTORAGTWN WHEN CLONING AN INSTANCE</td>
</tr>
<tr>
<td>1856040:</td>
<td>93=COPT-TO-DB-FILE OF WRKTRCLOG DOES NOT WORK</td>
</tr>
<tr>
<td>1800650:</td>
<td>ORA-9997 MCH6802 CPF9999 CPA0702 TRYING TO EXTEND GATEWAY TRACE MEMBER</td>
</tr>
<tr>
<td>1730629:</td>
<td>AGENT TRACE FILES NOT BEING PRODUCED WHEN JOB CCSID IS 930</td>
</tr>
<tr>
<td></td>
<td>Another DBCS problem</td>
</tr>
<tr>
<td>1714936:</td>
<td>ORA-28500 SQL0104</td>
</tr>
</tbody>
</table>
Known Restrictions

The restrictions documented in this section are known to exist for the products in this release of the Oracle Transparent Gateway for DB2/400. Refer to Chapter 9, "Developing Applications", for information about limitations when developing your applications.

Coexistence with Previous Version Gateway: Data Dictionary Tables

Once the data dictionary tables are installed by a release 9i Gateway, do not reinstall the data dictionary tables from a previous version Gateway, or you will not receive the benefits of the full release 9i capabilities.

Oracle Fast Refresh Snapshots

Oracle fast refresh snapshots are not supported between the gateway and the Oracle9i database server. However, Oracle complete refresh snapshots are supported between the gateway and the Oracle9i database server.

Oracle SQL Command INSERT

When copying data from an Oracle9i database server to a DB2/400 server, the Oracle SQL command INSERT is not supported. The SQL*Plus COPY command must be used. Refer to "SQL*Plus COPY Command" on page 7-13, for more information.

Verify the QCRTAUT Parameter Setting Before Installation

Before installing the gateway, verify that the QCRTAUT parameter is set to allow installation (not set to *EXCLUDE). Refer to "Installation Steps" on page 4-5 for information on verifying this value.

Date Arithmetic

Date arithmetic is not supported for SQL statements. For example, the following SQL expressions do not work:

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

Refer to Chapter 9, "Developing Applications" for additional information.
Julian Dates Support

You must use the J option on the TO_DATE and TO_CHAR functions to receive valid Julian dates.

GRAPHIC Constants in SQL Commands

GRAPHIC constants in SQL statements that are valid in one character set may not be valid as G-type or N-type GRAPHIC constants in DB2/400. For example, graphic constants in KO16KC5C5601 may contain both single-width and double-width characters. GRAPHIC constants in DB2/400 can contain only double-width characters.

WARNING: Oracle Corporation strongly recommends that you do not use graphic constants in SQL statements. If you do use graphic constants, then you should ensure that all of the characters within a graphic constant are double-width characters.

Oracle Corporation recommends that you use bind variables if you wish to use such constants. The gateway checks such bind variables (and the graphic constants in INSERT statements) and can force the single-width characters to become double-width characters if requested. With graphic data coming from a SELECT, the gateway can force those double-width characters back to single-width characters.

The ability to do this results in a data integrity exposure because the gateway cannot determine if a given double-width character was entered as a single-width character that was forced to become a double-width character.
Owners of DB2/400 Components

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

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 DB2/400 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 DB2/400 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, the SELECT statement fails.

GLOBAL_NAMES Initialization Parameter
If GLOBAL_NAMES is set to TRUE in the Oracle integrating server INIT.ORA file, then in order to be able to connect to the gateway, you must specify the Heterogeneous Services (HS) initialization parameter, HS_DB_DOMAIN, in the "Change Oracle Gateway Initialization Parameters" panel to match the value of the Oracle server DB_DOMAIN parameter. Refer to Chapter 6, "Configuring the Gateway" for more information.

Precompiler Limitations
The SQLCHECK option must be set to NONE when precompiling programs with the Oracle Precompilers.
Some SQL Functions Post-Processed

The Oracle9i database server provides more functionality for some SQL functions compared to similarly named DB2/400 SQL functions. As a result, these SQL functions are not passed through from the Oracle9i database server to DB2/400. They are, instead, post-processed. SUBSTR is an example of such a function. Some other SQL functions have different semantics in Oracle SQL and DB2/400 SQL. For details, refer to “SQL Functions” on page 9-23 in Chapter 9, "Developing Applications”.

DB2/400 SQL Limitations

Oracle ROWID Column

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

- UPDATE and DELETE are not supported with the WHERE CURRENT OF CURSOR clause. To update or delete a specific row through the gateway, a condition style WHERE clause must be used. (Bug No. 205538)
  
  When UPDATE and DELETE statements are used, in precompiler and PL/SQL programs, they rely internally on the Oracle ROWID function.

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

Oracle Bind Variables

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

For example, the following statements are not allowed:

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

CONNECT BY Is Not Supported

Oracle Transparent Gateway for DB2/400 does not support CONNECT BY in SELECT statements.
Oracle Fast Refresh Snapshots

Oracle fast refresh snapshots are not supported between the gateway and the Oracle9i database server. However, Oracle complete refresh snapshots are supported between the gateway and the Oracle9i database server.

Savepoints

If you try to issue a savepoint in DB2/400, you receive an ORA-2070 message. This error is appropriate because DB2/400 does not support savepoints.

LONG Datatypes

Any DB2/400 CHAR or VARCHAR column with a length greater than 4000 bytes is considered a LONG datatype by the gateway.

When using SELECT statements in SQL*Plus or any DB2/400 GRAPHIC or VARGRAPHIC column with a size greater than 2000 characters, the default size of the buffer that is used when retrieving LONG datatypes is 80. Therefore, if you wish to retrieve a DB2/400 CHAR or VARCHAR column with a length greater than 4000 bytes, then you must set the buffer length correctly. If the buffer size is not sufficient to contain the column value, then you will receive the following error message:

ORA-28528: Heterogeneous Services datatype conversion error

To work around this problem, reset the size of the buffer using the SET LONG command in SQL*PLUS or Server Manager.

In addition, if you use a program to retrieve a LONG column on the Oracle client, you must retrieve the entire LONG in one call to Oracle; you cannot retrieve the LONG in sections through the gateway.
This chapter describes the AS/400 hardware and software requirements for the Oracle Transparent Gateway for DB2/400. It contains the following sections:

- Hardware Requirements on page 3-2
- Software Requirements on page 3-3
- Documentation Requirements on page 3-3
- Distribution Kit on page 3-3
Hardware Requirements

Processor

The gateway requires any AS/400 processor that supports a version of OS/400 that is currently supported by IBM.

Memory

For most installations, 5 MB of virtual memory is recommended to support each active user of the Oracle Transparent Gateway for DB2/400.

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

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

CD-ROM Drive

A CD-ROM drive is required.
Disk Space

As distributed, disk space of 640 MB is required for installation of this product. Installation of further instances of the product at the same level use approximately 78 MB each.

These numbers are for initial conditions. The disk space can be reduced by running the REMOVEOBS procedure (refer to “Step 5: Remove Observability” on page 4-14). After running REMOVEOBS, each gateway instance will use 58 MB, and the common objects (service programs and the National Language files) will use approximately 60 MB.

Software 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 software vendors. Verify the latest support status with your system software vendors.

Operating System Requirements

This gateway requires any current version of OS/400 that is supported (with maintenance) by IBM.

Oracle Integrating Server

The Oracle server that is to act as the Oracle integrating server requires the latest released patch set for Oracle9i server or Oracle8i server.

Documentation Requirements

In addition to the manuals supporting the Oracle Transparent Gateway for DB2/400, the Oracle Net and Oracle9i database server manuals are recommended. You also need the appropriate AS/400 documentation for your system.

Distribution Kit

Before installing the gateway, verify that you have the correct CD-ROM and proper documentation:
Distribution Kit

- one Oracle Transparent Gateway for DB2/400 product CD-ROM
- appropriate Oracle documentation for installing, administering, and using the gateway: this book, *Oracle Transparent Gateway for DB2/400 Installation and User’s Guide*
- Oracle9i database server
Before you begin installation, be sure you have the required hardware and software described in Chapter 3, "System Requirements". This chapter contains the following sections:

- **Gateway Co-existence** on page 4-2
- **Checklists** on page 4-3
- **Preinstallation Steps** on page 4-11
- **Installation Steps** on page 4-11
- **Post-Installation Steps** on page 4-11
Gateway Co-existence

Oracle Transparent Gateway for DB2/400 release 9.2.0.1.0 can coexist with any previous version of the product. There are no known conflicts. Oracle Corporation does not support upgrading a previous version gateway to a version 9 gateway. You must reinstall. The various instances of the gateway on a machine have no knowledge of each other, other than through the DB2/400 files against which the gateways are executed, and through the data dictionary tables.

WARNING: After the data dictionary tables are installed by a Version 9 Gateway, do not reinstall the data dictionary tables from a Version 4 Gateway or a Version 8 gateway. If this is done by mistake, you can reinstall the Version 9 data dictionary table by using the CRTORADDB command that is described in Chapter 6, "Configuring the Gateway".
Checklists

Pre-Installation Checklist
- Step 1: Log on
- Step 2: System Value for the QCRTAUT Parameter

Installation Checklist
- Step 1: Mount the CD-ROM
- Step 2: Start the Install Process
- Step 3: Set Required Parameters
- Step 4: Verify Installation
- Step 5: Finish the Installation
- Step 6: Start the Gateway Subsystem

Post-Installation Checklist
- Step 1: Verify the Gateway Version
- Step 2: Verify Journalling
- Step 3: Journal the ORACLE2PC File
- Step 4: Configure Oracle Net
- Step 5: Remove Observability
- Step 6: Raise the Gateway Priority Level
Preinstallation Steps

Before installing the gateway, you must:

**Step 1: Log on**

Log on with a user profile that has *SECADM special authority; user profile QSECOFR has this authority.

**Step 2: System Value for the QCRTAUT Parameter**

Ensure that the system value for the QCRTAUT parameter is not *EXCLUDE. To find the current system value, enter the following from the command line:

```
WRKSYSVAL QCRTAUT
```

If the system value is *EXCLUDE, change it to any other less restrictive value to allow installation of the gateway. Remember to change QRCTAUT back to *EXCLUDE after installing the gateway.
Installation Steps

The panels below (beginning with Figure 4–1) are shown as they appear when first displayed. Any values on the panel are default values provided by Oracle Corporation. Default values are used unless you change them before going on to the next panel.

The installation procedure defines an OS/400 user ID that is identical to the name of the install library. This user ID is known as the gateway ID. The installation procedure also defines that ID with a user class of *USER, which has *JOBCTL authority. This ID is given authority over the entire library. To do that, the commands executed during installation are:

```
GRTOBJAUT OBJ(QSYS/gateway-id) OBJTYPE(*LIB) USER(gateway_id)
   AUT(*ALL)
GRTOBJAUT OBJ(gateway-id/*ALL) OBJTYPE(*ALL) USER(gateway_id)
   AUT(*ALL)
```

where `gateway-id` is both the gateway user ID and library name.

The security changes in the first six lines below may be made to allow the gateway to adopt the security authority of the "connect to" user ID. The GRTOBJAUT should be done only if the referenced user (in this case, *PUBLIC), does not already have the indicated authority to the object.

```
GRTOBJAUT OBJ(QSYS/QWTSETP) OBJTYPE(*PGM)
   USER(*PUBLIC) AUT(*USE)
GRTOBJAUT OBJ(QSYS/QSYGETPH) OBJTYPE(*PGM)
   USER(*PUBLIC) AUT(*USE)
GRTOBJAUT OBJ(QSYS/QSYSRLSPH) OBJTYPE(*PGM)
   USER(*PUBLIC) AUT(*USE)
```
You can use the EDTOBJAUT command to check the authorities of the three objects. For example, to check the current authority of the QWTSETP program in the QSYS library, use the following command:

EDTOBJAUT OBJ(QSYS/QWTSETP) OBJTYPE(*PGM)

If the *PUBLIC line shows *USE, then no change is needed.

Step 1: Mount the CD-ROM

Insert the CD-ROM into the AS/400 CD-ROM drive.

Step 2: Start the Install Process

Enter the command:

LODRUN drive

where drive is the name of the CD-ROM drive where you mounted the product CD-ROM. The panel in Figure 4–1 is displayed.

Step 3: Set Required Parameters

Panel
Install Oracle Transparent Gateway

Action
You must fill in the Instance name. Use the default name of ORACLE or enter a name from one to six characters long.

The installation procedure either creates a new library, uses a pre-existing, but empty, library, or uses a pre-existing library that has been created using the DB2/400 SQL command CREATE COLLECTION. If the library does not fit into this scenario, then it will not be used. Refer to Figure 4–1. The library must contain no objects other than those objects that were created by the CREATE COLLECTION command.

The preferred method is to use STRSQL to create a DB2/400 collection and then install the gateway into the library that is associated with that collection. Objects that are created later in the collection will be automatically journalled.
Figure 4–1  Install Oracle Transparent Gateway Panel

Install Oracle Transparent Gateway at 9.2.0.1.0

Type Choices, press Enter.

Instance name.............. ORACLE Name (up to six characters)

===>
F1=Help  F4=Prompt  F9=Retrieve  F10=Additional parameters  F12=Cancel

(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002

After entering a name, press [Enter]. The panel in Figure 4–2 appears.

Figure 4–2  Install Oracle Transparent Gateway Panel, With Name Choices

Install Oracle Transparent Gateway at 9.2.0.1.0
System: AS400A

Type Choices, press Enter.

Instance name.................. ORACLE Name (up to six characters)
Instance password.............. ORACLE 1-10 characters
Prestart jobs................... *YES *YES, *NO
TCP/IP port number............. 1521 1024-65534
Recovery user profile........... ORACLE Name
Recovery user password........... ORACLE 1-10 characters
Auxiliary storage pool id....... 1 1-16
Install Data Dictionary Support... *YSE *YES, *NO

===>
F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel

(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002

Installing the Gateway  4-7
You may change the choices:

**Table 4–1 Name Choices on Install Panel**

<table>
<thead>
<tr>
<th>Choice Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance name</td>
<td>is a name from one to six characters long. The default is ORACLE.</td>
</tr>
<tr>
<td>Instance password</td>
<td>is the password for the user ID (same as OS/400 User Profile) that is created during installation. The user ID has the same name as the instance name.</td>
</tr>
<tr>
<td>Prestart jobs</td>
<td>prestarts the TCP/IP jobs. Use *YES to prestart the TCP/IP jobs, or *NO to prestart NO jobs. You can change how many jobs are prestarted by changing the value of the initial number of TCP/IP jobs parameter by using the CHGORAPJE command. For more information, refer to “CHGORAPJE, Change Prestarted Job Parameters” on page 6-10.</td>
</tr>
<tr>
<td>TCP/IP port number</td>
<td>is 1521, which is the default. If port 1521 is unavailable, enter a different port number. Each instance requires a unique port number.</td>
</tr>
<tr>
<td>Recovery user profile</td>
<td>is a profile name you enter, or use the default of the gateway instance name.</td>
</tr>
<tr>
<td>Recovery user password</td>
<td>is a password for the recovery user profile. The default is the gateway instance name. This password must be kept in synchronization with the gateway password. Refer to “CHGRECOPRF, Change Recovery Profile Parameters” on page 6-16</td>
</tr>
<tr>
<td>Auxiliary storage pool id</td>
<td>uses the default of 1 or you can enter another ID.</td>
</tr>
<tr>
<td>Install Data Dictionary Support</td>
<td>turns on data dictionary installation. The default is YES. Installation of the data dictionary view support lengthens the installation process. There is only one Transparent Gateway data dictionary for the entire AS/400 machine. All Transparent Gateway instances use the same data dictionary. The data dictionary is created in the library OTGDB2.</td>
</tr>
</tbody>
</table>

After filling in the choices press [Enter] to continue.
Step 4: Verify Installation

Panel
Verify Oracle Install

Action
Enter *YES and press [Enter] to continue the installation process.

Figure 4–3 Verify Oracle Install Panel

Verify Oracle Install at 9.2.0.1.0

System: AS400A

Verify that you want to install the product. Once this process has started, it will create a library and other objects with the name of the instance specified below. Enter *YES to verify this operation before it is started.

New instance name ........... ORACLE
enter *YES to verify....... *NO *NO, *YES

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002
Step 5: Finish the Installation

If you entered *YES, a message displays on the screen about 5 to 15 minutes later to let you know the installation is finished. Check the job log for error messages. Use this command:

DSPJOBLOG

Step 6: Start the Gateway Subsystem

Enter the command:

STRSBS instance_name/instance_name

where instance_name is the name used in Step 3.

Installation of the gateway is complete.
Post-Installation Steps

Oracle Corporation recommends performing these post-installation steps. If you have more than one instance of the gateway, repeat the steps for each one. Note that you only need to install data dictionary support once. Refer to the table of installation choices in "Step 3: Set Required Parameters" for more information on data dictionary support.

Journalling

Journalling is the process of recording changes made to files on the AS/400 to ensure transactional consistency. Every table on the AS/400 that is going to be changed must be journalled. Changes to a file are recorded by a JOURNAL in a JOURNAL RECEIVER. Journalling is not required for gateways that are used for read-only purposes.

By default, the gateway runs under commitment control change (*CHG). This means that:
- AS/400 files not changed by SQL operations do not need to be journalled.
- Users can see changes input by others before the changes have been committed.

You can change the isolation level from the default setting, *CHG, to cursor stability (*CS). With cursor stability:
- Users cannot see current changes until they have been committed.
- All files changed by SQL operations must be journalled.

Refer to "CHGORATUN, Change Initialization Parameters" on page 6-10 for additional information about changing the isolation level.

For information about the *RR setting and more information about *CS and *CHG, refer to the Isolation Level section of the IBM reference for DB2/400.
Step 1: Verify the Gateway Version

To verify installation and find out what version of the gateway is running, use this command:

DSPPGM <instance_name>/GTW

where:

instance_name

is the name given to the gateway when it is installed.

GTW

is the name of the gateway program object (the gateway executable program).

The Text Description line resulting from the DSPPGM command informs you of the gateway version.

You can also find out what version of the gateway is running by looking in the job logs for the RUNORAGTTP job.

Step 2: Verify Journalling

Journal all AS/400 files changed through the gateway. To start journalling for a file, use the STRJRNPF command. To use the STRJRNPF command, you must have previously created a journal and a journal receiver, as illustrated in the following example. Note that if you installed the gateway into a library created as a SQL collection, then the journal and journal receiver already exist. You do not need to explicitly journal files that are created in a library that is part of a DB2/400 SQL collection. You can journal many files to the same pair of journal and journal receiver.

CRTJRNRCV JRNRCV(instance/QSQJRNRCV)
CRTJRN JRN(instance/QSQJRN) JRNRCV(instance/QSQJRNRCV)
STRJRNPF FILE(library/file) JRN(instance/QSQJRN)

where:

instance

is the name given to the gateway when it is installed.

library

is the name of the library where the journal is located.
file

is the name of the file you are journaling.

---

**Note:** If you attempt to change a file that is not journalled, you may receive an ORA-28500 error (connection from ORACLE to a non-Oracle system returned this message), followed by a second error message generated by the target system or gateway (usually a SQL 7008:filename in library not valid for this operation). In this case, enter CRTJRNRCV to create the journal receiver and then enter CRTJRN to create the journal. Then start journalling of the specified file by using the STRJRNPF command.

Refer to the Isolation Level section of the IBM reference for DB2/400 for more information about journalling and its parameters.

---

**Step 3: Journal the ORACLE2PC File**

This step is only required if:

- you are planning to issue a distributed update between the Oracle9i database server and the gateway
- any AS/400 stored procedures are executed

Journal the ORACLE2PC physical file, which records two-phase commit transactions. You must journal this file before issuing a distributed transaction to the AS/400. To journal the file enter:

```
STRJRNPF FILE(instance_name/ORACLE2PC) JRN(library/journal)
```

where:

- **instance_name**
  - is the name given to the gateway when it is installed.

- **library**
  - is the name of the library where the journal is located.

- **journal**
  - is the name of the journal.
**Step 4: Configure Oracle Net**


**Step 5: Remove Observability**

The gateway is shipped with IBM observability. Removing observability reduces gateway program object size by approximately 60 percent. If disk space is a concern, and if you have no plans to upgrade to a higher version of OS/400, then Oracle Corporation recommends that you remove observability from the gateway.

To remove observability, perform these steps:

1. Stop the gateway by entering:
   ```
   ENDSBS instance_name *IMMED
   ```
   where `instance_name` is the name given to the gateway when it is installed.

2. Use the SBMJOB command to submit a batch job using the QSECOFR ID. The SBMJOB syntax is:
   ```
   SBMJOB CMD(CALL instance_name/REMOVEOBS PARM(instance_name))
   ```

**Step 6: Raise the Gateway Priority Level**

The gateway subsystem and listener default to an AS/400 priority level of 50 during the installation process. In many cases, a priority level of 50 is conducive to performing the functions provided by the gateway. However, if the AS/400 is running at a high CPU capacity with several jobs competing for CPU allocation, it might be necessary to raise the priority of the gateway subsystem and listener to meet your performance requirements.

**Note:** The gateway runs as a batch job. Raising the priority can impact the performance of other processes. Raising the priority of the gateway higher than 20 is not recommended. (Lower numbers represent higher priority levels.)
Oracle Net is an Oracle product providing distributed database and processing capabilities. Generally, Oracle Net for OS/400 supports network communications between Oracle applications, Oracle servers, and Oracle gateways across different OS/400 systems or foreign operating systems. For product-specific information, refer to the Oracle9i Net Services Administrator’s Guide and the Oracle9i Net Services Reference Guide.

This chapter presents information about the Oracle Net architecture and how to configure and use Oracle Net for AS/400. It contains the following sections:

- **Overview of Oracle Net** on page 5-2
- **Checklists for Configuring Oracle Net** on page 5-7
- **Configuring Oracle Net for TCP/IP AS/400** on page 5-7
- **Connecting to the AS/400 through TCP/IP** on page 5-12
- **Troubleshooting TCP/IP Connection Problems** on page 5-15
Overview of Oracle Net

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

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

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

Distributed Processing

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

Distributed Database

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

The following terms are used to explain the architecture of Oracle Net for OS/400:

**host**

is the machine the database resides on and that runs the Oracle gateway. On OS/400, only a gateway can be running, the Oracle database does not run on the AS/400.

**client (task)**

is the application using a Oracle Net driver to communicate with the Oracle server or gateway. A server is also considered to be a client if it initiates a connection with another Oracle server, or with an Oracle gateway.

**protocol**

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

**driver**

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

**network**

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

Oracle Net for AS/400 Architecture

Oracle Net connections are established on the AS/400 through a listener. A listener receives incoming connections from Oracle Net clients and starts or transfers to a job on the AS/400 system.

On the AS/400, the Oracle Net TNS listener is used for TCP/IP.

**Figure 5–1** shows communication between a client and the AS/400. Oracle Net on the AS/400 can accept TCP/IP connections.
After the listener accepts the client connection, a batch job is started on the AS/400. The batch job executes the gateway and sends a response back to the client. The response is data or a message. Each batch job is responsible for a given client connection. One batch job is executed for a single connection, and a single connection is associated with only one batch job.

To reduce the waiting time for connecting to gateway jobs, a set of prestarted gateway batch jobs are associated with a TNS listener. The number of prestarted jobs and the point at which additional jobs are started can be modified by the system administrator.
Oracle Net File Name Structure

Oracle Net refers to files in the following format:

basename.extension

where basename is the base portion of the name, and extension is the second part of the name.

An example of this form is SQLNET.ORA.

On the AS/400, the Oracle Net parameter files are installed in the gateway instance library using the following file name mapping:

- the base portion of the file name is mapped to a member name in an OS/400 file
- the extension portion of the file name is mapped to an OS/400 file name

For example, the SQLNET.ORA parameter file is mapped to member SQLNET in the ORA file that is located in the gateway instance library. Member names are referred to as:

file(member_name)

Some of the parameter values in ORA(SQLNET) can be changed with the CHGORANET command. For more information, refer to “CHGORANET, Change Network Parameters” on page 6-8.

For a list of the network files and members that are provided with the gateway, refer to Appendix A, “Oracle Net Files and Members”.

Trace Files for Job Members

To enable tracing for the listener or server, use the CHGORANET command to change the values of the Listener trace level and Server trace level parameters. Possible trace level values are:

- *OFF (the default) provides no trace information. Tracing is not enabled.
- *USER provides the least detailed trace data
- *ADMIN provides more detailed trace data
- 16 provides comprehensive trace data.
Because trace files are large, you should use the least detailed trace level possible. For more information about changing the trace level parameters, refer to "CHGORANET, Change Network Parameters" on page 6-8.

When tracing is enabled, each job produces a trace file member in the TRC file with a member name in the format:

SERVxxxxxx  (for server jobs)

or

LISTxxxxxx  (for listener jobs)

where xxxxxx is the job number.

Note: You should follow the advice of Oracle Support Services for setting trace levels.
Checklists for Configuring Oracle Net

Checklist for Configuring Oracle Net for TCP/IP

- Step 1: Modify the Port Number
- Step 2: Define the Host Name
- Step 3: Verify the Host Name
- Step 4: Verify that the Listener is Started

Connecting through TCP/IP (on the Oracle Server) Checklist

- Step 1: Add a TCP/IP Connect Descriptor to TNSNAMES.ORA
- Step 2: Specify SID Names
Preliminary Step: Define a Physical Line

This step is required only if no physical connection currently exists between the Oracle9i database server and the AS/400.

To define the physical connection, use any of the following AS/400 commands:

- use CRTLINETH to create an ethernet connection.
- use CRTLINS DLC to create an SDLC connection.
- use CRTLINTRN to create a token ring connection.
- use CRTLINX25 to create an X.25 connection.

When you use one of these AS/400 commands, you can set the AUTOCREATE CONTROLLER parameter to *YES (the AS/400 system default is *NO). This enables the AS/400 auto-configuration feature.

If a line is already defined, then you can use the DSPLIND line_name command to display the line description parameters.

These line description parameters are used to configure the network on the machine where the gateway resides. For example, when using a token ring system, you must know the AS/400 token ring address.
Configuring Oracle Net for TCP/IP AS/400

When you install the gateway, you are prompted for a listener port, through which TCP/IP connections will be established. A listener job automatically starts when the gateway subsystem starts. The listener monitors the TCP/IP port you specified and accepts incoming connections directed to that port. After a connection is made, the listener starts a server job using the Oracle Net bequeath mechanism.

Internal process communication (IPC) between the listener and the server job is achieved through AS/400 local APPC devices. The listener then redirects the client connection to a randomly generated port assigned to the server job. This process is the Oracle Net inherit mechanism.

For more information about Oracle Net, refer to the Oracle9i Net Services Administrator’s Guide and the Oracle9i Net Services Reference Guide.

Figure 5–2 demonstrates a TCP/IP connection flow.

Figure 5–2  TCP/IP Connection Flow
Configuring for TCP/IP

Perform the following steps to configure your TCP/IP connection between the Oracle9i database server and your AS/400:

Step 1: Modify the Port Number
Step 2: Define the Host Name
Step 3: Verify the Host Name
Step 4: Verify that the Listener is Started

Step 1: Modify the Port Number

Many sites specify 1521 as the port number when the gateway is installed. If your gateway is already assigned to port 1521 or to another available port number, go to Step 2.

If your gateway is not assigned to a working port number, you must assign the gateway to a port it can use. Determine the number of an available port and assign it with the gateway CHGORANET command. You can find the values of port numbers currently active using the AS/400 command NETSTAT *CNN. Each running instance requires a unique listener port number; two gateway instances active at the same time cannot use the same listener port number. Refer to "CHGORANET, Change Network Parameters" on page 6-8 for more information.

Note: If you are using the AS/400 or TCP/IP for communications, refer to the IBM TCP/IP Fastpath manual (form SC41-5430) for information about TCP/IP concepts and terminology, setting up TCP/IP, and using TCP/IP. Correct configuration of TCP/IP on the AS/400 ensures that the gateway functions properly.
Step 2: Define the Host Name

To define the host name, perform these steps:

1. Run the AS/400 CFGTCP command.
2. Select option 12 Change TCP/IP domain information. The panel in Figure 5–3 appears:

Figure 5–3 Change TCP/IP Domain (CHGTCPDMN)

<table>
<thead>
<tr>
<th>Type Choices, press Enter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host name...</td>
</tr>
<tr>
<td>Domain name...</td>
</tr>
<tr>
<td>Domain search list...</td>
</tr>
<tr>
<td>Host name search priority...</td>
</tr>
<tr>
<td>Domain name server</td>
</tr>
<tr>
<td>Internet address...</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

3. Enter your domain name if the Domain name field is empty. Enter your host name if the Host name field is empty.

Press [Enter] to save your changes and return to the option list panel. (The changes take effect after you restart the gateway.)

The host name and the domain name combine to form the qualified host name. For example, AS400A combines with US.ORACLE.COM to form AS400A.US.ORACLE.COM as the qualified host name.
Step 3: Verify the Host Name

Verify your host name is in your host name table with the following AS/400 command:

PING host_name.domain_name

You can also use a PING command on the client-side Oracle database server to verify that it can communicate with the AS/400.

**Note:** The PING LOOPBACK command does not verify the host name.

Step 4: Verify that the Listener is Started

Use the Oracle Gateway Monitor to verify that the listener is started. If it is necessary to start the listener, use the Oracle Gateway Monitor to do so, or start the listener manually with this command:

```
STRORALSN
```

You can use the NETSTAT *CNN command to verify that the port that is assigned to the LISTENER is, in fact, being used by the LISTENER job in a given gateway instance. For more information on using the Oracle Gateway Monitor, refer to Chapter 8, "Administering the Gateway".

Connecting to the AS/400 through TCP/IP

Perform the following steps to configure your Oracle server machines for TCP/IP. Refer to the networking documentation for your Oracle server machine for more information about configuring it.

1. Step 1: Add a TCP/IP Connect Descriptor to TNSNAMES.ORA
2. Step 2: Specify SID Names
Step 1: Add a TCP/IP Connect Descriptor to TNSNAMES.ORA

Use the TCP/IP connect descriptor in the TNSNAMES.ORA file on the Oracle9i database server:

```plaintext
connect_name = (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCP)
        (PORT=1521)
        (HOST=host_name)
    )
    (CONNECT_DATA=(SID=csi_name))
    (HS=)
)
```

where:

- **connect_name**
  - is the name of the connection. This name must be unique within the TNSNAMES.ORA file. The `connect_name` corresponds to the value that is found with the USING keyword on a CREATE DATABASE LINK command.

- **TCP**
  - is the TCP protocol that is used for TCP/IP connections

- **1521**
  - is the default port number that is used by the Oracle Net listener on the AS/400. Change this value if you have changed the listener port number.

- **host_name**
  - is the TCP/IP hostname of your AS/400 machine.

- **csi_name**
  - is the name of the communication side information (*CSI) file that is used by the listener to start the server. Usually, the `csi_name` is ORAGATE. Refer to "Step 2: Specify SID Names" on page 5-14 for more SID names.

- **HS=**
  - This new keyword is mandatory. The right-hand side of the HS keyword is not needed.
Step 2: Specify SID Names

Oracle servers that are using TCP/IP to connect to a gateway must specify a system identifier (SID) name in the TNSNAMES.ORA file that is used to connect to the AS/400. The SID name is used on the AS/400 to start the server. The listener uses the SID name to reference a *CSI file. The *CSI file uses a TP_NAME to start the required CL program in the gateway server. Several different SID names are configured in the shipped system and are listed in the following table.

Set the SID to ORAGATE to use a prestarted job. ORAGATE would give the best response time. Setting the SID to the other values should be done in special situations as directed by Oracle Corporation support personnel.

The following table provides the name and usage of each SID:

**Table 5–1 SID Names That Are configured in the Shipped System**

<table>
<thead>
<tr>
<th>SID</th>
<th>*CSI</th>
<th>TP_NAME</th>
<th>CL Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORAGATE</td>
<td>ORAGATE</td>
<td>ORAGWTPJ</td>
<td>RUNORAGTTP</td>
</tr>
<tr>
<td>ORAGTPJ</td>
<td>ORAGTPJ</td>
<td>ORAGWTPJ</td>
<td>RUNORAGTTP</td>
</tr>
<tr>
<td>ORAGTEJ</td>
<td>ORAGTEJ</td>
<td>ORAGWTEJ</td>
<td>RUNORAGTTE</td>
</tr>
<tr>
<td>ORAGTTC</td>
<td>ORAGTTC</td>
<td>ORAGWTTTC</td>
<td>RUNORAGTTC</td>
</tr>
</tbody>
</table>

**Table 5–2 SID Names and Their Usage**

<table>
<thead>
<tr>
<th>SID</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORAGATE</td>
<td>uses one of the prestarted TCP/IP jobs.</td>
</tr>
<tr>
<td>ORAGTPJ</td>
<td>is an alias for ORAGATE.</td>
</tr>
<tr>
<td>ORAGTEJ</td>
<td>starts a new gateway job.</td>
</tr>
<tr>
<td>ORAGTTC</td>
<td>starts a gateway job and a CPIC trace. It is also used for debugging the connection between the listener and the server.</td>
</tr>
</tbody>
</table>

Refer to "Viewing Jobs" on page 7-7 for instructions on viewing jobs and a description of the jobs that typically exist.
Troubleshooting TCP/IP Connection Problems

If you cannot access AS/400 data using Oracle Net after configuring TCP/IP, check to see whether the AS/400 is reachable by pinging the AS/400 from an Oracle server host from the operating system prompt, or use a similar command from the shell prompt:

\texttt{PING host\_name.domain\_name}

where \texttt{host\_name} is the name identifying your AS/400.

If you cannot resolve your connection problem, refer to "Contacting Oracle Support Services" on page 10-4 for more information.

The following suggestions may resolve your connection problems:

\textbf{The AS/400 is not reachable:}

Review your Oracle Net configuration.

\textbf{The AS/400 is reachable, but you cannot access AS/400 data:}

Verify that the definition for the host name (from Step 2 of Configuring for TCP/IP) contains the fully qualified name. Without a fully qualified \texttt{host\_name.domain\_name}, Oracle Net cannot resolve the address, and you will be unable to access AS/400 data.

\section*{Resolving Connection Problems}

You may experience the following error types: ORA-28509, ORA-28511, ORA-28500, ORA-12154, or a hanging condition.

\textbf{ORA-28509}

\textbf{HS= is missing from your tnsnames.ora file:}

\texttt{HS=} is a new parameter that you must specify in the tnsnames.ora file, or you will not be able to use version 9 of the gateway. The correct syntax is shown at "Connecting to the AS/400 through TCP/IP" on page 5-12. If \texttt{HS=} is missing from your tnsnames.ora file, some symptoms are hanging connection and errors ORA-2068 and ORA-3114. Search Metalink for Note 136294.1, which provides more details about this topic.
**HS= not in correct place:**

HS= is specified in tnsnames.ora, but it is not specified in the correct place, or the number of parentheses is uneven (an odd number due to missing or extra parentheses).

HS= needs to be set up outside the CONNECT_DATA specification. If you misplace HS=, or if you do not have the correct parenthesis specification, then you can get error ORA-28509.

Other symptoms (same as when HS= is missing from your tnsnames.ora file) are hanging connection and errors ORA-2068 and ORA-3114.

**TCP/IP configuration:**

TCP/IP configuration under AS/400 (10"Option 10 of CFGTCP: Work with TCP/IP Host Table Entries" on page 5-17) is not configured as anticipated by the gateway listener.

The gateway listener requires \texttt{host\_name.domain\_name} in the TCP/IP host table entries. If you do not specify this, then you will not be able to get any response back from the gateway. You will be able to see the connection in the AS/400, but no data can be transferred.

**Gateway Listener is not up:**

The gateway listener is automatically started when you start the gateway instance. You can manually start the listener by executing the command: STRORALSN.

**ORA-28511**

**Gateway or Listener is not up:**

Check to see that the gateway is up and running correctly and is properly configured. Check to see if listener is up.

**Set Fully Qualified \texttt{host\_domain} Name**

Ensure that a fully qualified \texttt{host\_domain} name has been set on the AS/400. Use CFTTCP to do this. Use Option 10 on page 5-17 and Option 12 on page 5-18 to set a fully qualified \texttt{host\_domain} name. Use ping to verify that the \texttt{host\_domain} name matches the IP address of the host name.
ORA-28500

Check any accompanying message from the AS/400 to determine what the problem might be, or check the AS/400 job log (RUNORAGTTP job log). For example, message SQL-7008 indicates that the file is not journalled.

ORA-12154

This error indicates that a missing or incorrect tnsnames file has been specified, or that an incorrect spelling or service name has been specified in the tnsnames file. This is generally an Oracle Net issue, not a gateway issue.

Troubleshooting Your TCP/IP Configuration

You may encounter errors during the test phase of your gateway installation even though you are able to issue a TNSPING. If you do encounter such errors, please check the panels that were used to define the host name. Specifically, check the values presented when using option 12 of the CFGTCP command.

Option 10 of CFGTCP: Work with TCP/IP Host Table Entries

<table>
<thead>
<tr>
<th>Option</th>
<th>Internet Address</th>
<th>Host Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.5.40</td>
<td></td>
<td>AS400</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AS400.US.ORACLE.COM</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td></td>
<td>LOOPBACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOCALHOST</td>
</tr>
</tbody>
</table>

Enter Internet addresses and host names. Host names in the "Work with TCP/IP Host Table Entries" panel must include both the unqualified and the qualified names (the one that includes the domain name).

Option 10 depends upon Option 12. Option 12 defines your host name as well as your domain name. Under Option 12, you must enter your host name and your hostname.domain_name. This may seem redundant, but if the gateway is not configured correctly for TCP/IP, then the gateway will not make a connection, regardless of the output from the TNSPING command. Any changes to CFGTCP will require you to restart the gateway.

TCP configuration is documented in "Configuring Oracle Net for TCP/IP AS/400" on page 5-9 in this chapter.
Option 12 of CFGTCP: Change TCP/IP Domain (CHGTCPDMN)

<table>
<thead>
<tr>
<th>Option</th>
<th>Host name</th>
<th>Domain name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>’AS400A’</td>
<td>’US.ORACLE.COM’</td>
</tr>
</tbody>
</table>

The host name and the domain name combine to form the qualified host name. For example, AS/400A as illustrated in Option 12 combines with US.ORACLE.COM to form the qualified host name AS400A.US.ORACLE.COM as the qualified host name.
After installing the gateway, you can run gateway commands and change gateway parameters.

This chapter contains the following sections:

- Gateway Commands on page 6-2
- Setting Optional Parameters on page 6-18
Gateway Commands

All gateway parameters are changed with gateway commands, which are accessed through a menu system. These commands and their menus are described in this section.

The gateway comes with commands to do these tasks:

- copy (clone) the gateway
- change the most common gateway parameters
- change the level of tracing and debugging

All commands can be used after a gateway is installed. The following table summarizes each command and its purpose. The Menu Choice is used when executing the CMDORAGTW menu.

**Table 6–1 Command Summary**

<table>
<thead>
<tr>
<th>Menu Choice</th>
<th>Command Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CRTORAGTWI</td>
<td>creates a copy of an installed gateway version. You can copy a version as many times as needed. After making a copy, use the other gateway commands to change the parameters of the new copy.</td>
</tr>
<tr>
<td>2</td>
<td>CHGORANET</td>
<td>changes the values of network parameters.</td>
</tr>
<tr>
<td>3</td>
<td>CHGORAPJE</td>
<td>changes the parameter values of prestarted jobs.</td>
</tr>
<tr>
<td>4</td>
<td>CHGORATUN</td>
<td>changes the gateway initialization parameters.</td>
</tr>
<tr>
<td>5</td>
<td>CHGGTWDBG</td>
<td>sets or changes values for debugging parameters. (Use only under the guidance of a representative from Oracle Support Services.)</td>
</tr>
<tr>
<td>6</td>
<td>CRTORADDB</td>
<td>submits a batch job to create Oracle data dictionary views based on the DB2/400 system catalog.</td>
</tr>
<tr>
<td>7</td>
<td>CHGRECOPRF</td>
<td>changes the User Profile name, or the User Profile password, or both, for the User Profile that is associated with transaction recovery. For password precautions, refer to “CHGRECOPRF, Change Recovery Profile Parameters” on page 6-16.</td>
</tr>
</tbody>
</table>
Running the Commands

All gateway commands are accessed through a main menu. To invoke the main menu, enter:

```
ADDLIBLE  instance_name
GO CMDORAGTW
```

where: `instance_name` is the name given to the gateway when it was installed.

The main menu panel illustrated in Figure 6–1 is displayed:

**Figure 6–1  CMDORAGTW Oracle Commands Panel**

<table>
<thead>
<tr>
<th>CMDORAGTW</th>
<th>Oracle Commands</th>
<th>System: AS400A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select one of the following:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Create instance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Change network settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Change prestart job settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Change gateway initialization settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Change debugging options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Change Oracle Data Dictionary objects (batch)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Change Recovery Profile Parameters</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Selection or command

```plaintext
===>
```

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002
Gateway Commands

Enter the choice number (1, 2, 3, 4, 5, 6, or 7) or the corresponding command name:

- CRTORAGTWI
- CHGORANET
- CHGORAPJE
- CHGORATUN
- CHGGTWDBG
- CRTORADDB
- CHGRECOPRF

After you have made your selection, press [Enter] to continue. The panel for the command displays. Command panels are described under the individual commands.

Note: For more information about these choices, move the cursor to the value on the panel and press [PF1].

**CRTORAGTWI, Copy the Gateway**

You can have as many copies of a gateway instance on your system as you want. Once you have installed a version 9 gateway, if you need another instance of the gateway, before you issue the CRTORAGTWI command you must:

- Shut down the instance you are about to copy. Use the ENDSBS command with the instance name as the operand to perform the shutdown.
- Ensure you log on with a user profile that has *SECADM special authority; the user profile QSECOFR has this authority.
- Ensure the instance name to be created does not exist.

After entering 1 at the main menu panel or CRTORAGTWI, the panel in Figure 6–2 displays.
Enter the name of the new instance to be created and press [Enter] to continue. The panel in Figure 6–3 appears.
### Figure 6–3  Create Oracle Transparent Gateway Panel, with new values

<table>
<thead>
<tr>
<th>Create Oracle Transparent Gateway 9.2.0.1.0</th>
<th>System: AS400A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type choices, press Enter.</td>
<td></td>
</tr>
<tr>
<td>Existing instance name ........... ORACLE</td>
<td>Name (up to six characters)</td>
</tr>
<tr>
<td>Instance name to create ........... ORANEW</td>
<td>Name (up to six characters)</td>
</tr>
<tr>
<td>Instance password ............ ORANEW</td>
<td>1-10 characters</td>
</tr>
<tr>
<td>Recovery user profile ........ ORANEW</td>
<td>Name</td>
</tr>
<tr>
<td>Recovery user password ........ ORANEW</td>
<td>1-10 characters</td>
</tr>
<tr>
<td>Prestart jobs ................ *TCP/IP</td>
<td>*TCP/IP, *NONE</td>
</tr>
<tr>
<td>TCP/IP port number .............. 1521</td>
<td>1024-65534</td>
</tr>
<tr>
<td>Auxiliary storage pool id ....... 1</td>
<td>1-16</td>
</tr>
<tr>
<td>Install Data Dictionary           *NO</td>
<td>*YES, *NO</td>
</tr>
<tr>
<td>Support ....................... *NO</td>
<td></td>
</tr>
</tbody>
</table>

---

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel  
(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002

The first two entries are set from the previous screen and cannot be changed:

**Existing Instance name**

is a name from one to six characters long. The default is whatever is in the ORA_HOME data area. If you addlible `instance_name`, then ORA_HOME will be the instance name.

**Instance name to create**

is a name from one to six characters long.

You must fill in the following choices:

**Instance password**

is the password for the user ID that was created during installation. The user ID has the same name as the newly-created instance name. You should change this for security reasons.
Recovery user profile

is a profile name that you enter, or you can use the default of the newly-created gateway instance name.

Recovery user password

is a password for the recovery user profile. The default is the newly-created gateway instance name. You should change this for security reasons. Use the OS/400 CHGUSRPRF command to make the change. Then, use the CHGRECOPRF command (refer to "CHGRECOPRF, Change Recovery Profile Parameters" on page 6-16).

Prestart jobs

prestarts the TCP/IP jobs. By default, two TCP/IP jobs are prestarted. Use *TCP/IP to prestart the TCP/IP jobs. Use *NONE to prestart no jobs.

You can change how many jobs are prestarted by changing the value of the initial number of TCP/IP jobs parameter with the CHGORAPJE command. For more information, refer to "CHGORAPJE, Change Prestarted Job Parameters" on page 6-10.

TCP/IP port number

is 1521, which is the default. If port 1521 is unavailable, enter a different port number. Each instance requires a unique port number.

Auxiliary storage pool id

uses the default of 1, or you can enter another ID if you have additional auxiliary storage pools defined.

Install Data Dictionary Support

turns data dictionary view support on or off. The default is *NO. Installation of the data dictionary view support lengthens the installation process. If you are copying a gateway, the Data Dictionary was most likely installed when the copied gateway itself was installed. In that case, you do not need to reinstall it.
CHGORANET, Change Network Parameters

After entering 2 at the main menu panel or CHGORANET, enter the appropriate instance name and press [Enter]. The panel in Figure 6–4 appears. Except when you are changing the value for the TCP/IP port number, you should use CHGORANET only under the guidance of a representative from Oracle Support Services. Enter the new values and press [Enter] to continue. The new values do not take effect until you shut down and restart the gateway that was specified in the Existing instance name parameter of the command.

Figure 6–4  Change Oracle Network Parameters Panel

<table>
<thead>
<tr>
<th>Change Oracle Transparent Gateway</th>
<th>System: AS400A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type choices, press Enter.</td>
<td></td>
</tr>
<tr>
<td>Existing instance name ..........</td>
<td>ORACLE Name</td>
</tr>
<tr>
<td>TCP/IP port number .............</td>
<td>1521 1024-65534</td>
</tr>
<tr>
<td>Client trace level .............</td>
<td>*OFF *OFF, *USER, *ADMIN, 16</td>
</tr>
<tr>
<td>Listener trace level ...........</td>
<td>*OFF *OFF, *USER, *ADMIN, 16</td>
</tr>
<tr>
<td>Server trace level .............</td>
<td>*OFF *OFF, *USER, *ADMIN, 16</td>
</tr>
<tr>
<td>Regenerate files ...............</td>
<td>*NO *YES, *NO</td>
</tr>
</tbody>
</table>

===>

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002
Changing the ORA(SQLNET_CHG) File

Some network parameters are documented in the ORA(SQLNET) file but are not displayed on the CHGORANET panel. These parameters cannot be changed directly by editing the ORA(SQLNET) file. They must be changed in the ORA(SQLNET_CHG) file:

1. Use an OS/400 editor to change parameter values in ORA(SQLNET_CHG).
2. On the OS/400 command line, use the CHGORANET command, or use the GO CMDORAGTW command and specify option 2.
3. Enter the instance name of the gateway, and press Enter.
4. Specify *YES on the "Regenerate Files" line, and press Enter.

The ORA(SQLNET) file now reflects the values specified in the ORA(SQLNET_CHG) file.

Changing the ORA(LISTEN_CHG) File

There are network parameters documented in the ORA(LISTENER) file that are not displayed on the CHGORANET panel. These parameters cannot be changed directly by editing the ORA(LISTENER) file. They must be changed using the ORA(LISTEN_CHG) file:

1. Use an OS/400 editor to change parameter values in ORA(LISTEN_CHG).
2. On the OS/400 command line, use the CHGORANET command, or use the GO CMDORAGTW command and specify option 2.
3. Enter the instance name of the gateway, and press Enter.
4. Specify *YES on the "Regenerate Files" line, and press Enter.

The ORA(LISTENER) file now reflects the values specified in the ORA(LISTEN_CHG) file.
CHGORAPJE, Change Prestarted Job Parameters

On the OS/400 command line, use the CHGORAPJE command, or use the GO CMDORAGTW command, and specify option 3, and press Enter. The panel in Figure 6–5 appears. Enter the new values and press [Enter] to continue.

Figure 6–5  Create Oracle Prestart Parameters Panel

The new values do not take effect until you shut down and restart the gateway specified in the Existing instance name parameter of the command.

CHGORATUN, Change Initialization Parameters

On the OS/400 command line, use the CHGORATUN command, or use the GO CMDORAGTW command and specify option 4. Then press Enter. The panel in Figure 6–6 appears. CHGORATUN uses two panels. Refer to Figure 6–6 and Figure 6–7.
Note: CHGORATUN can be run only while signed on as the user profile corresponding to the instance ID, or as a user profile that has *SECADM special authority (QSECOFR has such authority). Furthermore, the value specified for "User Profile CCSID" can be changed only by the user profile corresponding to the instance ID (the instance ID is the name given to the library in which the gateway was installed).

For information on what values you should enter for these parameters, move the cursor to the value on the panel and press [PF1].

Enter the new values on the panels. Before entering values for the DATABASE DOMAIN, GATEWAY NATIONAL LANGUAGE, RPC FETCH REBLOCKING, and RPC FETCH SIZE gateway initialization parameters, read the following notes:

Table 6–2 Initialization Parameter Notes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notes</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE DOMAIN</td>
<td>If the value for the GLOBAL_NAME parameter is set to TRUE in the Oracle9i database server INIT.ORA file, then the value you enter in this field must match that specified for the DB_DOMAIN parameter in the Oracle9i database server INIT.ORA file.</td>
<td>WORLD</td>
</tr>
<tr>
<td>Gateway Language</td>
<td>The GATEWAY NATIONAL LANGUAGE line no longer appears in the CHGORATUN screen, but NLS_NCHAR actually still does exist for other reasons.</td>
<td>AMERICAN_AMERICA.WE8EBCDIC37</td>
</tr>
<tr>
<td>RPC FETCH REBLOCKING and</td>
<td>If the RPC FETCH REBLOCKING parameter is set to YES (the default), then the block size of the buffer for SELECT statements is determined by the value of the RPC FETCH SIZE parameter. The recommended value for Oracle Transparent Gateway for DB2/400 is 40,000. The RPC FETCH SIZE parameter defines the maximum number of bytes sent with each fetch between the gateway and the Oracle9i database server. Each fetch between the gateway and the Oracle9i database server can contain multiple rows from DB2/400.</td>
<td>YES 40 000</td>
</tr>
<tr>
<td>RPC FETCH SIZE</td>
<td>V4 GRAPHIC compatibility mode and UCS-2 support Before deciding on a value for this parameter, refer to &quot;DB2/400 GRAPHIC Support&quot; on page 9-17 for more information.</td>
<td>NO</td>
</tr>
</tbody>
</table>

Configuring the Gateway 6-11
Figure 6–6  Change Oracle Gateway Initialization Parameters (first page of panel)

Change Oracle Gateway Initialization Parameters

Type choices, press Enter.

Existing instance name .......... ORACLE
Database Domain ................ WORLD
Database Name .................. ORACLE
Array block size ................. 100
Gateway language ............... american_america.e8ebcdic37

V4 Graphic and UCS-2
   Compatibility mode .......... *NO

Maximum Date ....................
Option for CCSID=65535 fields  *BITDATA
User Profile CCSID ............... *SYSVAL

===>

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel

(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002
**Figure 6–7  Change Oracle Gateway Initialization Parameters (second page of panel)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Isolation Level</td>
<td>*CHG</td>
<td>*CHG, *CS, *RR</td>
</tr>
<tr>
<td>Set gateway for READ-ONLY</td>
<td>*NO</td>
<td>*YES, *NO</td>
</tr>
<tr>
<td>Maximum Number Cursors</td>
<td>200</td>
<td>50–200</td>
</tr>
<tr>
<td>RPC Fetch Reblocking</td>
<td>*YES</td>
<td>*YES, *NO</td>
</tr>
<tr>
<td>RPC Fetch Size</td>
<td>40000</td>
<td>4000–50000</td>
</tr>
</tbody>
</table>

The new values do not take effect until you shut down and restart the gateway specified in the Existing instance name parameter of the command.
CHGGTWDBG, Change Debugging Parameters

This command is used for isolating the cause of a suspected gateway problem. Only use CHGGTWDBG under the guidance of a representative from Oracle Support Services.

On the OS/400 command line, use the CHGGTWDBG command, or use the GO CMDORAGTW command and specify option 5. then press [Enter]. Enter the appropriate instance name and press [Enter]. The panel in Figure 6–8 appears. For more information about the value choices, move the cursor to the value on the panel and press [PF1].

> Figure 6–8 Change Oracle Gateway Debugging Option Panel

After entering the new values, press Enter to continue. The new values do not take effect until you shut down and restart the gateway specified in the Existing instance name parameter of the command.
CRTORADD, Create Data Dictionary Views

This command submits a batch job to create Oracle data dictionary views of the DB2/400 system catalog. If you are going to run an application such as Developer 2000, you need to create data dictionary views. Therefore, if you create these views when originally installing the gateway, you should not need to create them again.

On the OS/400 command line, use the CRTORADD command, or use the GO CMDORAGTW command and specify option 6. Then press [Enter]. Enter the appropriate instance name and press [Enter]. A batch job is submitted. No additional panel appears.

The Oracle Data Dictionary views of the DB2/400 system catalog are used by all Oracle Gateway instances on that AS/400. A single copy of these views is on each AS/400.
CHGRECOPRF, Change Recovery Profile Parameters

This command changes the OS/400 User Profile name or password, or both, for the User Profile name or password that the gateway uses when directed to perform transaction recovery by the Oracle server. Transaction recovery is necessary when any failure occurs during a distributed transaction. The User Profile name and password are created by the OS/400 CRTUSRPRF command or are changed by the OS/400 CHGUSRPRF command.

Figure 6–9  Change Recovery Profile Parameters Panel

<table>
<thead>
<tr>
<th>Change Oracle Recovery Profile Parameters</th>
<th>System: AS400A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type choices, press Enter.</td>
<td></td>
</tr>
<tr>
<td>Existing instance name . . . . ORACLE</td>
<td>Name</td>
</tr>
<tr>
<td>Recovery Profile Name . . . ORACLE</td>
<td>Profile Name</td>
</tr>
<tr>
<td>Recovery Profile Password . . . Password</td>
<td></td>
</tr>
<tr>
<td>Confirmation of Recovery</td>
<td></td>
</tr>
<tr>
<td>Profile Password . . . . . . Password</td>
<td></td>
</tr>
</tbody>
</table>

F1=Help  F4=Prompt  F9=Retrieve  F12=Cancel
(C) COPYRIGHT ORACLE CORPORATION, 1994, 2002

Enter the User Profile name in the "Recovery Profile Name" field, or leave it as it is. Enter the password for the User Profile name on the next two lines. The password is stored in an encoded form. Because this password is no longer visible in plain text, extra care may be required to keep the recovery password synchronized between the AS/400 computer and the gateway. Refer to the following note.
**Caution:** You must keep the gateway recovery password synchronized with the AS/400. The gateway recovery user ID and password must be valid to the AS/400 at all times. If the recovery password is not valid (if it expires, for example), then an in-doubt transaction occurs, and the Oracle database alert log receives error-1017 (invalid user ID or password). Use the CHGRECOPRF command to change the recovery profile parameters (including recovery user ID and password).
Setting Optional Parameters

You can change the values of optional gateway parameters after the product is installed by using the gateway commands. Three commonly changed parameters are:

- **Array block size** in the BLOCKSIZE data area. Refer to "Retrieving Data" for more information.
- **Default character conversion** in the ORARAW data area. Refer to "Data Conversion" for more information.
- **Set gateway for READ-ONLY** for configuring the gateway with read-only capabilities. Refer to "Read-Only Gateway" for more information.

Retrieving Data

The gateway can retrieve multiple rows from a DB2/400 table or view with a single fetch. The gateway uses the BLOCKSIZE data area to determine the number of rows to retrieve. These conditions apply for the BLOCKSIZE data area:

- If the BLOCKSIZE data area is set to 0, no block retrieval is performed. This is the same as setting BLOCKSIZE to 1.
- If the BLOCKSIZE data area is set to \( n \), the gateway retrieves \( n \) rows in a single fetch, where \( n \) is a value from 1 to 32767.
- If the BLOCKSIZE data area does not exist, the gateway will retrieve one row per fetch to DB2/400.

**Note:** For performance reasons, Oracle Corporation recommends the BLOCKSIZE data area be set between 10 and 100.

The gateway uses a default value of 100 for the BLOCKSIZE data area. The BLOCKSIZE value is used for substitution for host variable :N in a SQL FETCH statement such as:

```
FETCH CN for :N ROWS...
```

To change this default value, use the gateway command CHGORATUN. After displaying the panel for CHGORATUN, enter a new value for Array block size. Refer to "CHGORATUN, Change Initialization Parameters" on page 6-10 for information about changing the setting for Array block size.
Data Conversion

The default coded character set identifier (system value QCCSID) for the AS/400 system is 65535. This CCSID value indicates to the gateway that character data in a column with such a CCSID is not to be converted and is to be treated as bit data.

The line entitled "Option for CCSID=65535 fields" on the "Change Oracle Gateways parameters" panel (use the CHGORATUN command) specifies how the gateway is to handle the "For Bit Data" and "CCSID=65535" fields. If the specification is *BITDATA, then the fields are treated as binary data and no translation occurs. If *CHARDATA is specified then the fields are treated as if they were in the character set ID in which the gateway runs. When using the CHGORATUN command, the "User Profile CCSID" line specifies the character set ID in which the gateway runs. The ORARAW date area is used to hold the data conversion specification.

Read-Only Gateway

The gateway can be configured with read-only capabilities. The read-only option may provide improved performance and security based on your configuration and parameter selections. The READONLY data area controls whether the gateway is enabled in this mode. The default setting for the read-only feature is NO. You can change the value of this environment parameter using the CHGORATUN command.

If you enable the read-only feature by changing the setting to YES, only queries (SELECT statements) are allowed to DB2/400. The capabilities which control whether updates are allowed through the gateway are disabled. These capabilities include insert, update, delete, and stored procedure support (pass-through SQL, DB2/400 stored procedures). Statements attempting to modify records at DB2/400 are rejected.

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

Refer to "CHGORATUN, Change Initialization Parameters" on page 6-10 for information about changing the READONLY setting.
After installing the gateway, you can administer database links, access the gateway, access AS/400 file members, perform distributed queries, and copy data between the Oracle9i database server and the AS/400.

This chapter contains the following sections:

- Database Link Behavior on page 7-2
- Using DB2/400 Cursors on page 7-5
- Using the Synonym Feature on page 7-5
- Accessing the Gateway on page 7-6
- Accessing AS/400 File Members on page 7-7
- Performing Distributed Queries on page 7-8
- Replicating in a Heterogeneous Environment on page 7-11
- Copying Data from the Oracle9i database server to the DB2/400 Server on page 7-12
- Copying Data to the Oracle9i database server from the DB2/400 Server on page 7-14
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 Oracle9i 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 DB2/400 database.

Connections to the DB2/400 database might be limited by factors that include memory, gateway parameters, or DB2/400 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 Reference. Information for using database links with the gateway is discussed here.

Creating Database Links

To create a database link in Oracle and to define a path to the gateway, use the CREATE DATABASE LINK statement. The CONNECT TO clause specifies the remote user ID and password to use when creating a session in the gateway. If you do not specify a user ID and password in the CONNECT TO clause, then the Oracle logon user ID 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 DB2/400 database using Oracle Net:

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

where:

- **dblink** is the complete database link name (such as gateway).
- **userid** is the user ID used to establish a session in the remote database. It must be authorized to any table or file on the DB2/400 server referenced in the SQL commands. The user ID cannot be longer than ten characters.
- **password** is the password used to establish a session in the remote database. This must be a valid DB2/400 server password. The password cannot be longer than ten characters.
- **tns_name_entry** specifies the Oracle Net TNS connect descriptor used to identify the gateway subsystem. This is identical to the CONNECT_NAME found in entries in the TNSNAMES.ORA member. Refer to "Step 1: Add a TCP/IP Connect Descriptor to TNSNAMES.ORA" on page 5-13.

Closing 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

DB2/400 tables, views, and synonyms available to the user ID specified in the CONNECT TO clause can be accessed with the following syntax:

```
SELECT * FROM table@gateway
```

or

```
SELECT * FROM user.table@gateway
```

The CONNECT TO user ID provides implicit qualification for unqualified tables. For example:

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

resolves to SCOTT.EMP on DB2/400 if the CONNECT TO user is SCOTT. If no CONNECT TO statement is defined with the database link, then the Oracle user ID using the database link is used as the implicit qualifier.

Oracle Corporation strongly recommends that your table or view specifications (or both) be always qualified with the table or view owner, as in the following example:

```
SELECT * from user.table@gateway.
```

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 PUBLIC DATABASE LINK dblink;
```

Do not drop a database link if it might be required to resolve an in-doubt distributed transaction. Refer to the Oracle9i Administrator’s Reference 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
Using the Synonym Feature

- You can provide complete data, location, and network transparency by using the synonym feature of the Oracle9i 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 DB2/400 table. Refer to the Oracle9i database server documentation for details on the synonym feature.

- The following statement creates a system-wide synonym named EMPDB2 in the Oracle database for the SCOTT.EMP file in the DB2/400 server:

  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
Accessing the Gateway

To access the gateway, complete the following steps on the Oracle9i database server:

1. Login to the Oracle9i database server.
2. Create a database link to the AS/400 database with the following syntax:

   ```sql
   CREATE DATABASE LINK AS400
   CONNECT TO userid IDENTIFIED BY password
   USING 'as400'
   ```

3. Retrieve data from the AS/400 database using one of these methods:
   - if the CONNECT TO clause of the database link specified ORACLE as the user ID, then this query retrieves data from the EMP file in the ORACLE library, using the name ORACLE as the AS/400 user profile:
     ```sql
     SELECT * FROM EMP@AS400
     ```
   - this query retrieves the EMP file in the library or collection CORPDATA, using the name ORACLE as the AS/400 user profile. The ORACLE AS/400 user profile must have the appropriate AS/400 privileges to access the CORPDATA.EMP file:
     ```sql
     SELECT * FROM CORPDATA.EMP@AS400
     ```

   These messages are displayed if insufficient privileges were granted to Oracle:
   ```text
   ERROR at line 1:
   ORA-28500: Connection from Oracle to non-Oracle system returned this message
   SQL0551: Not authorized to object EMP in CORPDATA type *FILE
   ORA-02063: preceding 2 lines from AS400
   ```

4. Update data on the AS/400.

5. Before attempting to update a file on the AS/400, ensure the file is currently being journalled. If the file is not journalled, these messages are displayed:

   ```text
   ORA-28500: Connection from Oracle to non-Oracle system returned this message
   SQL7008: EMP in CORPDATA not valid for operation.
   ORA-02063: preceding 2 lines from AS400
   ```
**Viewing Jobs**

Use the WRKACTJOB command to view all jobs. If the gateway job is active, then you will see the "Listener" job in SELW (select wait) status on the WRKACTJOB panel under the gateway subsystem. When the listener attempts to "hand-off" a new connection to a pre-started job, its status is CPCW (wait for completion of CPI communications call).

From the WRKACTJOB panel, press PF14 to see all the RUNORAGT* jobs. Two RUNORAGTTP jobs are usually present, both in PSRW (pre-start, or ready to run) status.

If active gateway connection exists, then the RUNORAGT* jobs can be in either RUN or TIMW (time wait) status.

**Accessing AS/400 File Members**

An AS/400 physical file can have multiple members. However, the AS/400 does not allow you to directly access a physical file member through SQL. For example, if you have an AS/400 file named EMP in the library CORPDATA with members MBR1, MBR2, and MBR3, issuing the SQL statement:

```
SELECT * FROM CORPDATA.EMP.MBR1
```

results in an error.

If your AS/400 file has multiple members, you can use AS/400 logical files to point to some or all of the members. To access a file member, create a logical file over the physical file members. In the physical file data members parameter (DTAMBRS) of the CRTLF command, specify which members you want to access. Using the previous example, if you wanted to access members MBR1 and MBR2 of EMP, you would enter the following command:

```
CRTLF FILE(CORPDATA/EMPLF) DTAMBRS((CORPDATA/EMP (MBR1 MBR2)))
```

This creates a logical file, EMPLF, that points to MBR1 and MBR2 of the EMP file. The SQL statement:

```
SELECT * FROM CORPDATA.EMPLF
```

returns all the rows that are contained in MBR1 and MBR2.
You can also specify DTAMBRS(*ALL) to include all members of a physical file in a logical file.

---

**Note:** To insert into a specific member, you must create a logical file for that member.

---

### Accessing Flat Files

Flat files are AS/400 files that are created by a utility other than DB2/400. All flat files have an externally described attribute. The externally described attribute refers to whether or not a file is defined by an external template. All DB2/400 files are externally described. You can access externally described flat files through the gateway in the same way that you access DB2/400 file members. You cannot access flat files that are not externally described. Files in the Integrated File System are not externally defined and are therefore not accessible to DB2/400.

The syntax for accessing flat files is:

```
library.file
```

where:

- `library` is the library name.
- `file` is the file member (flat file) name.

If flat files have multiple members, you need to create a view for each member or combination of members.

### Performing Distributed Queries

The gateway technology can execute distributed queries that join the Oracle9i database server and the AS/400, and any other data store for which Oracle Corporation provides a gateway. These complex operations can be completely invisible to the users requesting the data.

### Example of a Distributed Query

```
SQL command 1
```

This example joins data between the Oracle database server and the AS/400:
SELECT O.CUSTNAME, P.PROJNO, E.ENAME, SUM(E.RATE*P.HOURS)
FROM ORDERS@DB2 O, EMP@ORACLESRVR E, PROJECTS@AS400 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, the process of distributed queries may become invisible to the user. For example:

CREATE SYNONYM ORDERS FOR ORDERS@DB2
CREATE SYNONYM PROJECTS FOR PROJECTS@AS400
CREATE VIEW DETAILS (CUSTNAME,PROJNO,ENAME,SPEND)
AS
SELECT O.CUSTNAME, P.PROJNO, E.ENAME, SUM(E.RATE*P.HOURS)
FROM ORDERS O, EMP@ORACLESRVR E, PROJECTS P
WHERE O.PROJNO = P.PROJNO
AND P.EMPNO = E.EMPNO
GROUP BY O.CUSTNAME, P.PROJNO, E.ENAME

SQL command 2

Using the following SQL statement, a user can retrieve information from these three data stores in a single simple command:

SELECT * FROM DETAILS;

The results from SQL command 2 are the same as those from SQL command 1, and might look like the following:

<table>
<thead>
<tr>
<th>CUSTNAME</th>
<th>PROJNO</th>
<th>ENAME</th>
<th>SPEND</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC Co.</td>
<td>1</td>
<td>Jones</td>
<td>400</td>
</tr>
<tr>
<td>ABC Co.</td>
<td>1</td>
<td>Smith</td>
<td>180</td>
</tr>
<tr>
<td>XYZ Inc.</td>
<td>2</td>
<td>Jones</td>
<td>400</td>
</tr>
<tr>
<td>XYZ Inc.</td>
<td>2</td>
<td>Smith</td>
<td>180</td>
</tr>
</tbody>
</table>

Two-Phase Commit Processing

For a database to fully participate in a two-phase commit transaction, it must support both a prepare phase and a commit phase for committing transactions. The prepare phase ensures all participating nodes referenced in a distributed transaction are prepared to commit or abort the transaction, regardless of intervening failures.
The Oracle9i database server supports two-phase commit transactions. Therefore, any number of Oracle database servers can participate in a distributed two-phase commit transaction. The prepare phase is performed when a COMMIT is issued at the end of a distributed transaction.

**Note:** The prepare phase occurs automatically when an application COMMIT is executed. No other action is necessary.

DB2/400 does not support a prepare phase for committing a transaction. Consequently, it does not support two-phase commit transactions. Therefore, the two-phase commit protocol is limited when the gateway participates in a distributed transaction. In this case, the gateway becomes the commit point site of the distributed transaction. Because the gateway is configured as commit and confirm, it is always the commit point site, regardless of the COMMIT_POINT_STRENGTH setting of any of the participating Oracle databases. The gateway commits the local AS/400 unit of work after verifying all Oracle databases in the transaction have successfully committed their work.

Because the gateway must drive the distributed transaction, only one gateway can participate in an Oracle two-phase commit transaction.

Two-phase commit transactions are recorded in the ORACLE2PC physical file. You must journal this file before issuing a distributed transaction to the AS/400. Refer to "Step 3: Journal the ORACLE2PC File" on page 4-13 for more information.

For additional information about the two-phase commit process, refer to Oracle9i Database Administrator’s Guide.
Recovering Failed Transactions

If a two-phase commit transaction fails because the database connection is lost, the pending transaction is stored on the Oracle9i database server. Every time a user logs in to the gateway, the system checks to see if the tables on the Oracle9i database server that point to the DB2/400 link contain pending transactions. If so, the gateway suspends the login request until the recovery process is complete.

During the recovery process, the server uses the recovery user ID and password to login to the system and resolve pending transactions. After recovery is complete, the gateway executes the user’s login request. This recovery process is transparent to the user.

The recovery user ID and password are initially set in the “Install Oracle Transparent Gateway Panel, With Name Choices”, Figure 4–2. If you need to change these values, then you must run the CHGRECOPRF command. Refer to “Step 3: Set Required Parameters” on page 4-6 for more information.

Replicating in a Heterogeneous Environment

Oracle Transparent Gateway for DB2/400 provides a number of options for replicating Oracle and non-Oracle data throughout the enterprise.

Oracle9i database server Triggers

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

Oracle Snapshots

Oracle Transparent Gateway for DB2/400 can use the Oracle snapshot feature to automatically replicate non-Oracle data into the Oracle9i 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 Oracle9i database server to the DB2/400 Server

Data can be copied from the Oracle database server to the DB2/400 server by two methods:

- Triggers
- SQL*Plus COPY command

Triggers

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

For example, suppose you have a Oracle table ORA_EMP that contains fields ENAME and EMPNO. You also have a table called DB2_EMP, which is a copy of ORA_EMP and which resides on DB2/400. You want all changes made to the Oracle ENAME field to be reflected immediately in your DB2_EMP table on DB2/400. In the following scenario, an Oracle database server trigger can be developed to run every time an update is made to the ENAME field 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 that is used to access the gateway.
**SQL*Plus COPY Command**

The SQL*Plus COPY command copies data from the Oracle9i database server to the DB2/400 server. The SQL command INSERT is not supported as a way to copy tables from the Oracle server to the gateway. The following command:

```
INSERT INTO gateway_table@gateway 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 DB2/400 server:

```
COPY FROM username/password@ORACLESRV R -
  INSERT destination_table@gateway -
  USING query;
```

The next example selects all rows from the local Oracle EMP table and inserts them into the EMP table on the DB2/400 server:

```
COPY FROM SCOTT/TIGER@ORACLESRV R -
  INSERT SCOTT.EMP@gateway -
  USING SELECT * FROM EMP;
```

**Note:** Even though the SQL*Plus COPY command supports APPEND, CREATE, INSERT, and REPLACE options, INSERT is the only option supported when copying to the DB2/400 server. For more information about the COPY command, refer to the *SQL*Plus User’s Guide and Reference.
Copying Data to the Oracle9i database server from the DB2/400 Server

Use one of the following options to copy data from the DB2/400 server to the Oracle9i database server:

- Use the CREATE TABLE command to copy data from the DB2/400 server to the Oracle9i database server. To create a table on your local database and insert rows from a DB2/400 table, use:

  
  ```sql
  CREATE TABLE table_name AS query;
  ```

  The next example creates the table EMP in the local Oracle9i database server and inserts the rows from the EMP table on the DB2/400 server:

  ```sql
  CREATE TABLE EMP AS SELECT * FROM SCOTT.EMP@gateway;
  ```

- Use the INSERT command to copy data from the DB2/400 server to the Oracle9i database server:

  ```sql
  INSERT INTO oracle_table SELECT * FROM db2table@gateway;
  ```

  The following example selects all rows from the EMP table on the DB2/400 server and inserts them into the local Oracle EMP table:

  ```sql
  INSERT INTO EMP SELECT * FROM SCOTT.EMP@gateway;
  ```

- Use the CREATE SNAPSHOT command to automatically and asynchronously copy DB2/400 server data into the Oracle9i 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 SQL Reference. To create a copy:

  ```sql
  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 AS/400 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 DB2/400 server to the Oracle database server:

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

where `username` or `password`, or both, are valid at the Oracle server that is identified by `tns_alias`, and `query` refers to a remote table that is accessed via a gateway database link.

The following example selects all rows from the EMP table in DB2/400 and inserts them into the local Oracle EMP table:

```
COPY FROM SCOTT/TIGER@ORA920 -
INSERT EMP -
USING SELECT * FROM SCOTT.EMP@gateway;
```
Read this chapter to understand how to administer your gateway either by using the Oracle Gateway Monitor or by entering commands at the command line prompt. It contains the following sections:

- Overview of the Oracle Gateway Monitor on page 8-2
- Starting the Oracle Gateway Monitor on page 8-2
- Starting and Stopping the Gateway on page 8-5
- Starting the TCP/IP Listener on page 8-6
- Displaying the Gateway Call Stack on page 8-6
- Working with Gateway Job Locks on page 8-6
- Displaying Open Files on page 8-6
- Gateway Security on page 8-6
Overview of the Oracle Gateway Monitor

The Oracle Gateway Monitor is a facility provided with the Oracle Transparent Gateway for DB2/400 that allows a single interface to perform these functions:

- start and stop an Oracle gateway instance
- work with Oracle gateway jobs
- start the TCP/IP listener job
- display the gateway call stack
- work with the gateway job locks
- display the open files of the gateway

Starting the Oracle Gateway Monitor

In order to execute the Oracle gateway monitor, you must log on to the gateway instance user ID and have the gateway instance library included in your library list.

To start the Oracle gateway monitor, enter this command:

```
ORAMON
```

The main menu for the Oracle Gateway Monitor is displayed as shown in Figure 8–1.
Figure 8–1  Monitor Oracle Subsystems

Monitor Oracle Subsystems

System:  AS400A

Position to _____________  Update frequency  *OFF

Type Options, press Enter.

2=Start  4=End  5=Work with  7=Listen  10=Display call stack

11=Work with locks  14=Display open files

<table>
<thead>
<tr>
<th>Option</th>
<th>Name</th>
<th>Type</th>
<th>Status</th>
<th>Job</th>
<th>Aux</th>
<th>Aux</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>ORACLE</td>
<td>SBS</td>
<td>ACTV</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>LISTENER</td>
<td>LIS</td>
<td>SELW</td>
<td>015826</td>
<td>564</td>
<td>3386</td>
<td>2.8</td>
</tr>
<tr>
<td>-</td>
<td>RUNORAGTP</td>
<td>PJ</td>
<td>PSRW</td>
<td>015823</td>
<td>124</td>
<td>3725</td>
<td>1.6</td>
</tr>
<tr>
<td>-</td>
<td>RUNORAGTTP</td>
<td>PJ</td>
<td>PSRW</td>
<td>015822</td>
<td>138</td>
<td>3737</td>
<td>1.5</td>
</tr>
</tbody>
</table>

F1=Help  F2=Change list  F3=Exit  F5=Refresh  F8=Start auto update

The columns in the panel of Figure 8–1 represent the following:

Option is used to perform various functions on subsystems and jobs. Enter an option number next to the subsystem or job name and then press ENTER. You can select options for more than one subsystem or job at a time. The option numbers are listed in Figure 8–1.

Name displays the name(s) of Oracle subsystems and jobs.

Type is the type of entry for this line:

ASJ- automatically started job
LIS- listener
PJ- pre-started job waiting for start request
SBS- subsystem
SRV- server job
UNK- unknown
Starting the Oracle Gateway Monitor

To display specific gateway subsystems and jobs, you must enter the subsystem into the Oracle gateway monitor subsystem by pressing [PF2]. This displays a panel that allows you to enter the name of the Oracle gateway subsystems. The panel in Figure 8–1 shows a single gateway subsystem with a single listener and two prestarted gateway jobs.

To work with a specific gateway subsystem or job, you must enter an option in the Option field next to the subsystem or job displayed on the main panel.

In Figure 8–1 the Update frequency field is turned off. This means that no updates of the information are displayed until you press [PF5]. If you want the screen to be updated periodically, you need to press [PF8]. The default update refresh is 30 seconds, but you can set the value between 30 and 600 seconds.

If help is needed at any time, pressing [PF1] on any of the panels displays more information about a specific panel.
Starting and Stopping the Gateway

Entering a 2 in the Option field on the main panel allows you to start a gateway subsystem or job. ACTV displays in the STATUS field when the gateway subsystem is started. In order to start the gateway subsystem, you must be authorized to use the STRSBS command.

To stop the gateway subsystem, use the Option field next to the subsystem or job you want to shut down and enter a 4. This ends the gateway subsystem or job specified. When the gateway subsystem has shut down, INACTV displays in the field next to the subsystem or job that ended. You must be authorized to use the ENDSBS command to shut down the gateway.

Working with Oracle Gateway Jobs

To work with gateway jobs, specify option 5 next to the job. This displays the AS/400 Work with Jobs menu to work with a specific gateway job.

Starting and Stopping the Gateway Using the Command Line

An instance of the Oracle Transparent Gateway for DB2/400 runs within an OS/400 subsystem. All jobs relating to that instance run within the subsystem. You use the STRSBS command to start a subsystem. For example, to start an instance of a gateway called ORACLE, you enter the following command:

```
STRSBS ORACLE/ORACLE
```

where the first ORACLE is a library name and the second ORACLE is a subsystem description.

To stop a gateway you must end the subsystem using the ENDSBS command. For example, to stop a gateway called ORACLE and immediately end all jobs within the ORACLE subsystem, and stop the subsystem itself, you enter the following command:

```
ENDSBS ORACLE OPTION(*IMMED)
```
Starting the TCP/IP Listener

Specify option 7 in the Option field next to the gateway subsystem name if you want to start the gateway listener. The gateway listener is identified by LIS in the TYPE field displayed on the main menu.

Displaying the Gateway Call Stack

Displaying the gateway call stack is not needed for most installations, but might be requested by Oracle Support Services during problem isolation and determination. Entering option 10 in the Option field displays a call stack for a specific job.

Working with Gateway Job Locks

Specify option 11 in the Option field next to a specific job to display and work with object locks held by the gateway. This option displays the AS/400 Work with Job Locks panel.

Displaying Open Files

Specify option 14 in the Option field next to a specific job to show the files open for that specific job. This option displays the AS/400 Display Open Files panel.

Gateway Security

The Oracle user ID and password are passed over the database link to the gateway to authorize gateway users to DB2/400 objects. If the CONNECT TO clause is specified when creating the database link, then the user ID and password that are 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 user ID and password that are using the database link are passed to the gateway for authorization.
Caution: You must keep the gateway recovery password synchronized with the AS/400. The gateway recovery user ID & password must be valid to the AS/400 at all times. If the recovery password is not valid (if it expires, for example), then an in-doubt transaction occurs, and the Oracle database alert log receives error-1017 (invalid user ID or password). The CHGRECOPRF command is used to change the recovery profile parameters (including recovery user ID and password).

For more information about using CHGRECOPRF, refer to "CHGRECOPRF, Change Recovery Profile Parameters" on page 6-16.
Oracle Transparent Gateway for DB2/400 allows applications written for the Oracle Server to access tables in a DB2/400 database. Using a database link, the access can be made transparent by using synonyms or views of the DB2/400 tables. However, there are fundamental SQL, datatype, and semantic differences between the Oracle server and DB2/400. Read this chapter to learn these differences and for information on developing applications.

To develop applications that access AS/400 data through the gateway, you need to understand the topics in this chapter. It contains the following sections:

- **Gateway Appearance to Application Programs** on page 9-2
- **Array Processing** on page 9-3
- **Using Oracle Stored Procedures with the Gateway** on page 9-5
- **Using DB2/400 Stored Procedures with the Gateway** on page 9-7
- **Passing DB2/400 SQL Statements through the Gateway** on page 9-11
- **Converting DB2/400 Datatypes to Oracle Datatypes** on page 9-15
- **SQL Functions** on page 9-23
- **Oracle Server SQL Construct Processing** on page 9-25
- **Oracle Server and DB2/400 Differences** on page 9-27
- **Oracle Data Dictionary Emulation in a DB2/400 Server** on page 9-28
- **Oracle Developer Forms Compatibility** on page 9-29
Gateway Appearance to Application Programs

An application written to access information in a DB2/400 database interfaces with an Oracle server. When developing applications, remember the following:

- You must define the DB2/400 database to the application by use of a database link. Your application specifies tables existing on a DB2/400 database using the name defined in the database link. For example, if you define a database link naming the DB2/400 database link as DB2400, and an application needs to retrieve data from an Oracle database and the DB2/400 database, the following SQL statement retrieves data from both Oracle and DB2/400:

  ```sql
  SELECT EMP.EMPNO, EMPS.SALARY FROM EMP, EMPS@DB2400
  WHERE EMP.EMPNO = EMPS.EMPNO;
  ```

  In this example, EMP is a table on the Oracle server and EMPS is a table on the DB2/400 server. Alternatively, you can define a synonym or a view referencing the DB2/400 server table and access the information without the database link suffix.

  For example:

  ```sql
  CREATE SYNONYM EMPS FOR EMPS@DB2400;
  SELECT EMP.EMPNO, EMPS.SALARY FROM EMP, EMPS
  WHERE EMP.EMPNO = EMPS.EMPNO;
  ```

- You can perform reads and writes of data to a defined DB2/400 database. SELECT, INSERT, UPDATE, and DELETE are all valid operations.

- A single transaction can NOT write to more than one DB2/400 database. A stored procedure call via DB2/400 counts as a write.

- Single SQL statements, using a JOIN, can refer to tables in multiple Oracle databases, multiple DB2/400 databases, or both.

Connecting to DB2/400 Databases with Database Links

A connection to the gateway is established through a database link when that link is first used in an Oracle server session. The connection remains active until the session ends. Another session or user can access the same database link but will get a separate connection to DB2/400.

Connections to DB2/400 might be limited by factors such as memory, gateway parameters, or AS/400 server resources.
Read and Write Access to DB2/400 Databases

You can perform read and write functions to a DB2/400 database. SELECT, INSERT, UPDATE and DELETE are all valid operations.

A single transaction can use an UPDATE statement to change multiple Oracle databases. Only one DB2/400 database can be updated within a transaction. In addition, only one DB2/400 database may participate in a stored procedure call.

A single transaction can use a SELECT statement to retrieve data from multiple Oracle databases and multiple DB2/400 databases.

The gateway is setup with commitment control set to *CHG. Refer to the IBM reference for IBM DB2 for OS/400 DB2/400SQL for more information regarding AS/400 commitment control.

Accessing Tables in Multiple Databases

Single SQL statements, using JOINs, can refer to tables in multiple Oracle databases or multiple DB2/400 databases.

Array Processing

When evaluating and tuning your gateway configuration, you may achieve performance gains by using Oracle's 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 DB2/400. 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 may save elapsed time and CPU cycles. In addition, when using INSERT for multiple rows, DB2/400 processing is optimized by retaining the original SQL statement for repeated execution.

The array size is a factor in transferring data between the client application (for example, SQL*Plus) and the Oracle integrating server when using UPDATE, DELETE, and INSERT statements, in that it allows one UPDATE, DELETE or INSERT statement to be used with a series of values. For more information on array processing usage and implementation in your Oracle application, refer to Oracle9i SQL Reference.
Fetch Reblocking

The array size of the application for SELECT is effective between the application and the Oracle9i database server. However, the block size of the buffer between the integrating Oracle server and the gateway is controlled by two initialization parameters: RPC FETCH SIZE and RPC FETCH REBLOCKING. These gateway initialization parameters correspond to the Heterogeneous Services initialization parameters HS_RPC_FETCH_SIZE and HS_RPC_FETCH_REBLOCKING respectively. Figure 9–2 below illustrates this. Refer to Oracle9i Database Administrator’s Guide for more information.

With version 9 of the gateway, the Oracle9i database server supports fetch reblocking with the RPC FETCH REBLOCKING parameter. When the value of this parameter is set to *YES (the default), the block size of the buffer for SELECT statements is determined by the RPC FETCH SIZE value. The RPC FETCH SIZE parameter defines the number of bytes sent with each buffer from the gateway to the Oracle9i database server. The buffer might contain one or more qualifying rows from DB2/400. This feature can provide significant performance enhancements, depending on your application design, installation type, and workload. The RPC FETCH REBLOCKING and RPC FETCH SIZE parameters may be changed by using the CHGORATUN command. Refer to “CHGORATUN, Change Initialization Parameters” on page 6-10 for more information.

Note: For performance reasons, Oracle Corporation recommends setting the initial Oracle application array size between 10 and 100.
The array size between the client and the Oracle9i database server is still determined by the Oracle application. To ensure optimal performance, Oracle Corporation recommends that you set your Oracle application array size equal to the gateway’s ARRAY BLOCK SIZE parameter. For more information on the gateway’s ARRAY BLOCK SIZE parameter, refer to "Setting Optional Parameters" on page 6-18. For more information on array processing usage and implementation in your Oracle application, refer to Oracle9i SQL Reference.

Figure 9–2 Array Size Definition in the DB2/400 Architecture for SELECT

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 continual use. Applications use standard Oracle PL/SQL to invoke stored procedures.

Oracle stored procedures can be located in a local instance of Oracle and a remote instance. Figure 9–3 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.
If the application wants to maintain location transparency, a synonym can be created:

```sql
CREATE SYNONYM ORAPROC2 FOR oraproc2@ora2;
```

where `oraproc2` is the procedure stored in ORA2.

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

In Figure 9–3, 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 DB2/400 tables through the gateway.

In Figure 9–4, `empproc` is an Oracle stored procedure that subsequently accesses data in DB2 using the Oracle Transparent Gateway for DB2/400.
Like the Oracle gateway, standard PL/SQL is used to create and execute the procedure. There is no difference in the gateways, except that the stored procedure is accessing DB2/400 instead of Oracle.

Gateway two-phase commit processing also applies to updates to DB2/400 being made within an Oracle stored procedure. This means that the stored procedure can update a single instance of DB2/400 while also updating any number of Oracle databases within a single transaction.

A call to a DB2/400 stored procedure counts as a write since, as far as the Oracle Transparent Gateway for DB2/400 is concerned, the gateway cannot tell if the stored procedure is actually changing anything on the AS/400. Therefore, you cannot have a transaction that calls two DB2/400 stored procedures via two separate database links. Correspondingly, during a transaction, you cannot call a DB2/400 stored procedure via one database link and also do an INSERT, UPDATE, or DELETE relative to DB2/400 using another database link (even if the two database links resolve to the same AS/400).

Using DB2/400 Stored Procedures with the Gateway

The procedural feature of the gateway enables execution of native DB2/400 stored procedures. In other words, the stored procedure is no longer defined in the Oracle database, but instead is defined to DB2/400. Again, standard Oracle PL/SQL is used by the Oracle application to execute the DB2/400 stored procedure.
The gateway does not require special definitions in order to invoke the DB2/400 stored procedure. Once the stored procedure is defined to DB2/400, the gateway is able to use the existing DB2/400 definition to execute the procedure.

In Figure 9–5, an Oracle application calls the empproc stored procedure that is defined to DB2/400.

From the perspective of the application, executing the DB2/400 stored procedure is no different than invoking a stored procedure at a remote Oracle instance.

### Executing DB2/400 Stored Procedures from Applications

In order for an application to invoke a DB2/400 stored procedure, it is first necessary to create the DB2/400 stored procedure on the DB2/400 system using the procedures documented in the IBM reference for DB2/400.

After the stored procedure is defined to DB2/400, 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 DB2/400 stored procedure REVISE_SALARY. The DB2/400 stored procedure retrieves the salary value from the DB2/400 database to calculate a new yearly salary for JOHN SMYTHE.
The revised salary returned in RESULT is used to update EMP in a table of an Oracle database:

```
DECLARE
    INPUT VARCHAR2(15);
    RESULT NUMBER(8,2);
BEGIN
    INPUT := 'JOHN SMYTHE';
    MYLIB.REVISE_SALARY@DB2400 (INPUT, RESULT);
    UPDATE EMP SET SAL = RESULT WHERE ENAME := INPUT;
END;
```

When the gateway receives a call to execute a DB2/400 stored procedure, it first does a lookup of the procedure in the QSYS2/SYSPROCS and QSYS2/SYSPARMS DB2/400 tables to determine:

1. The stored procedure to execute.

   The gateway retrieves information from the ROUTINE_SCHEMA and ROUTINE_NAME columns of the QSYS2/SYSPROCS table to locate the DB2/400 stored procedure to be invoked by the gateway. Multiple DB2/400 stored procedures can have the same ROUTINE_NAME, but must be uniquely identified by the combination of the ROUTINE_SCHEMA and ROUTINE_NAME columns of the QSYS2/SYSPROCS table.

   The gateway always receives a qualified stored procedure name from the Oracle server and executes the DB2/400 stored procedure using this qualified name. If a qualified name is not explicitly identified by the application, then the user ID that is passed over the database link is used to match the value in ROUTINE_SCHEMA. To ensure that you execute the correct DB2/400 stored procedure, Oracle Corporation recommends you use fully qualified stored procedure names when invoking DB2/400 stored procedures from Oracle PL/SQL.

2. The parameter list of the stored procedure.

   When a DB2/400 stored procedure is defined, the attributes of each entry in the argument list for the stored procedure are placed in the QSYS2/SYSPARMS table. The gateway uses this information to convert the datatype of each argument to its corresponding PL/SQL datatype. Refer to "Converting DB2/400 Datatypes to Oracle Datatypes" on page 9-15 for gateway datatype conversion rules.

After the lookup of the description of the DB2/400 stored procedure is complete, the gateway uses this information to execute the DB2/400 stored procedure.
Considerations for Using Stored Procedures with DB2/400

There are special considerations for using the procedural feature with the Oracle Transparent Gateway for DB2/400:

- PL/SQL records cannot be passed as parameters when invoking a DB2/400 stored procedure.

- The gateway executes the DB2/400 CALL statement to execute DB2/400 stored procedures. Commitment control (of database changes that are made with the DB2/400 stored procedure) is managed by DB2/400. Refer to the IBM reference for DB2/400 for more information regarding commitment control.

- The gateway supports both GENERAL and GENERAL WITH NULLS linkage conventions of DB2/400 stored procedures.
  - The GENERAL linkage convention means that the parameters passed to and from DB2/400 stored procedures cannot be null.
  - The GENERAL WITH NULLS linkage convention means that parameters passed to and from DB2/400 stored procedures can be null when they are passed using indicator variables.

Oracle Call Interface (OCI) or embedded PL/SQL can be used in host programs to operate on indicator variables. For information on writing DB2/400 stored procedures to use indicator variables, refer to the IBM reference for DB2/400.

- To ensure commitment control integrity between the gateway and the stored procedure program, you must create the *PGM object containing the DB2/400 stored procedure using the activation group attribute *CALLER. If you do not use the *CALLER attribute, stored procedures execute in a different activation group than the gateway. If the activation group attribute was given as *NEW, then a forced COMMIT may occur when the activation group ends (that is, when the program object defined by the stored procedure ends). For more information on activation groups, refer to the IBM document for ILE concepts.

- If the stored procedure does not change any file or any DB2/400 table on the AS/400, then the program object may be an OPM program object. Otherwise, the program object must be an ILE program object. Failure to adhere to this may cause data to become unsynchronized in the case of a ROLLBACK that either may be planned by you or may be forced because of some failure.
Passing DB2/400 SQL Statements through the Gateway

The passthrough SQL feature allows an application developer to send a SQL statement directly to DB2/400 without the statement being interpreted by the Oracle9i database server. DB2/400 SQL statements that are 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 DB2/400 SQL statements using the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function.

DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE is a function that is built into the gateway. This function takes 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 is a reserved name of the gateway and is used specifically for running native DB2/400 SQL.

This release of Oracle Transparent Gateway for DB2/400 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 9-14 for more information.

Using the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE Function

To execute a passthrough SQL statement using the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function, use the following syntax:

```
number_of_rows = DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink('DB2/400_sql');
```

where:

- `number_of_rows` is a variable that is assigned the number of rows that are affected by the passthrough SQL execution. For DDL statements, a zero is returned for the number of rows affected.
- `dblink` is the name of the database link that is used to access the gateway.
Examples

Following are examples of the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function.

Refer to "Using Passthrough Statements to Create Tables" on page 9-13 for more information.

1. Insert a row into a DB2/400 table using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE:

   ```sql
   declare
   num_rows NUMBER;
   begin
   num_rows := DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink
   ('INSERT INTO SCOTT.DEPT VALUES ( 10, ''PURCHASING'',
       ''PHOENIX'')');
   end;
   /
   ```

2. Create a DB2/400 table using DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE:

   ```sql
   declare
   num_rows NUMBER;
   begin
   num_rows := DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@dblink
   ('CREATE TABLE MYTABLE (COL1 INTEGER, COL2 INTEGER, COL3
       CHAR(14),
       COL4 VARCHAR(13))');
   end;
   /
   ```

DB2/400_sql is a valid DB2/400 nonquery SQL statement (except CONNECT, COMMIT, or ROLLBACK). DB2/400 SQL statements that cannot be dynamically prepared are rejected by DB2/400. The SQL statement that is passed by the DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE function must be a character string. Refer to the IBM reference for DB2/400 for more information regarding DB2/400 SQL statements.
Using Passthrough Statements to Create Tables

If you use DBMS_HS_PASSTHROUGH.EXECUTE.IMMEDIATE to create a DB2/400 table, then one of the following is required for journalling:

- a QSQJRN journal in the target library
- a target library that was created as a collection

(This requirement applies if the DB2/400 isolation level is set to *CHG, which is the default.)

If neither of the above is true, then you will receive an error message that the table has been created but cannot be journalled. In which case, you can disregard the error message and journal the file manually. Refer to "Step 2: Verify Journalling" on page 4-12 and "Step 3: Journal the ORACLE2PC File" on page 4-13 for more information on journalling.
Retrieving Results Sets Through Passthrough

Oracle Transparent Gateway for DB2/400 provides a facility to retrieve results sets from a SELECT SQL statement issued through passthrough. In the example below the values for a single field are selected from all the rows of the foreign table PT_TABLE, and these values are inserted into the local Oracle table PT_TABLE_LOCAL. Refer to Oracle9i Database Administrator’s Guide for additional information.

Example

```sql
DECLARE
  CRS binary_integer;
  RET binary_integer;
  VAL VARCHAR2(10)
BEGIN
  CRS:=DBMS_HS_PASSTHROUGH.OPEN_CURSOR@gtwlink;
  DBMS_HS_PASSTHROUGH.PARSE@gtwlink(CRS,'SELECT NAME FROM PT_TABLE');
  WHILE (TRUE)
    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;
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('END OF FETCH');
    DBMS_HS_PASSTHROUGH.CLOSE_CURSOR@gtwlink(CRS);
  END;
END;
/```
Converting DB2/400 Datatypes to Oracle Datatypes

To move data between applications and the underlying database, the gateway binds data values from a host variable or literal to a datatype that is understood by the underlying database.

Oracle tools expect Oracle datatypes. Consequently, the gateway maps values from DB2/400 servers into appropriate Oracle datatypes before passing these values back to the application or Oracle tool. The following table lists the datatype mapping and restrictions:

<table>
<thead>
<tr>
<th>DB2/400</th>
<th>Oracle</th>
<th>Criteria</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR(N)</td>
<td>CHAR(N)</td>
<td>1 ≤ N ≤ 255</td>
<td>Maximum length of a DB2/400 CHAR column is 32766 characters.</td>
</tr>
<tr>
<td></td>
<td>VARCHAR2(N)</td>
<td>255 &lt; N ≤ 4000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LONG</td>
<td>4000 &lt; N</td>
<td></td>
</tr>
<tr>
<td>VARCHAR(N)</td>
<td>VARCHAR2(N)</td>
<td>N ≤ 4000</td>
<td>Maximum length of a DB2/400 VARCHAR column is 32740 characters.</td>
</tr>
<tr>
<td></td>
<td>LONG</td>
<td>4000 &lt; N ≤ 32740</td>
<td></td>
</tr>
<tr>
<td>LONG VARCHAR(N)</td>
<td>VARCHAR2(N)</td>
<td>N ≤ 4000</td>
<td>Maximum length of a DB2/400 VARCHAR column is 32740 characters.</td>
</tr>
<tr>
<td></td>
<td>LONG</td>
<td>4000 &lt; N ≤ 32740</td>
<td></td>
</tr>
<tr>
<td>VARCHAR(N) for Bit Data</td>
<td>RAW(N)</td>
<td>1 ≤ N ≤ 255</td>
<td>Maximum length of a DB2/400 VARCHAR column is 32740 characters.</td>
</tr>
<tr>
<td></td>
<td>LONG RAW(N)</td>
<td>255 &lt; N ≤ 32740</td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Refer to “Performing Date and Time Operations” on page 9-20.</td>
<td>Refer to “Performing Date and Time Operations” on page 9-20.</td>
</tr>
<tr>
<td>TIME</td>
<td>CHAR(8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>CHAR(26)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2/400</td>
<td>Oracle</td>
<td>Criteria</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>* GRAPHIC(N)</td>
<td>CHAR(2*N)</td>
<td>1 ≤ N ≤ 127</td>
<td>Maximum length of a DB2/400 GRAPHIC column is 16370 graphic characters</td>
</tr>
<tr>
<td></td>
<td>VARCHAR2(2*N)</td>
<td>127 &lt; N ≤ 16370</td>
<td></td>
</tr>
<tr>
<td>* VARGRAPHIC(N)</td>
<td>VARCHAR2(2*N)</td>
<td>1 ≤ N ≤ 2000</td>
<td>Maximum length of a DB2/400 VARGRAPHIC column is 16370 graphic characters</td>
</tr>
<tr>
<td></td>
<td>LONG(2*N)</td>
<td>2000 &lt; N</td>
<td></td>
</tr>
<tr>
<td>* LONG VARGRAPHIC(N)</td>
<td>VARCHAR2(2*N)</td>
<td>1 ≤ N ≤ 2000</td>
<td>Maximum length of a DB2/400 VARGRAPHIC column is 16370 graphic characters</td>
</tr>
<tr>
<td></td>
<td>LONG(2*N)</td>
<td>2000 &lt; N</td>
<td></td>
</tr>
<tr>
<td>FLOAT(N)</td>
<td>FLOAT(21)</td>
<td>1 ≤ N ≤ 21</td>
<td></td>
</tr>
<tr>
<td>FLOAT(N)</td>
<td>FLOAT(53)</td>
<td>22 ≤ N ≤ 53</td>
<td></td>
</tr>
<tr>
<td>DECIMAL(P,S)</td>
<td>NUMBER(P,S)</td>
<td></td>
<td>Packed decimal on the AS/400</td>
</tr>
<tr>
<td>NUMERIC(P,S)</td>
<td>NUMBER(P,S)</td>
<td></td>
<td>Zoned decimal on the AS/400</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NUMBER(10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMALLINT</td>
<td>NUMBER(5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Table Note: If the column CCSID is 13488 (UCS-2), and if the gateway NLS_LANG specifies a single-byte character set, then replace "2*N" in the Oracle column with "N", replace 127 with 255, and replace 2000 with 4000. If the column CCSID is 13488 and Oracle is expecting single-byte data, then (VAR)GRAPHICs are treated as (VAR)CHARs.
DB2/400 GRAPHIC Support

Oracle8i server supports NCHAR and NVARCHAR2 datatypes in order to support multi-byte character sets. Oracle9i server does not support NCHAR or NVARCHAR2 datatypes. With Oracle9i, the Oracle Transparent Gateway for DB2/400 supports double-byte datatypes (GRAPHIC and VARGRAPHIC in DB2/400 terms) in a manner similar to the manner in which they were supported in version 4 of the Oracle Transparent Gateway for DB2/400. That is to say, in Oracle tools and applications, they are supported as CHAR and VARCHAR2 datatypes of the appropriate size.

In order to support DB2/400 columns of GRAPHIC or VARGRAPHIC type, you must set NLS_LANG to a language in which the character set part corresponds to the CCSID of the (VAR)GRAPHIC column. Thus, if the column is CCSID 933, then the character set part of NLS_LANG must be set to KO16DBCS (refer to Appendix B, "National Language Support"). In addition, you must specify on the CHGORATUN screen that you want V4 graphics compatibility, and the gateway job should be running in the CCSID of the GRAPHICs columns. The job CCSID is also set on the CHGORATUN panel.

For DBCS Users

Double-Byte Character Support

This version of the Oracle Transparent Gateway for DB2/400 allows you to access and change columns in DB2/400 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 a SQL0105 error: "Mixed or Graphic String Constant Not Valid."

An ASCII client might attempt to use the following INSERT statement:

```
INSERT INTO mytable.graphcol@tg4db2400 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 DB2/400 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@tg4db2400 values('AxxB')
```

If you use the FORCE_SB option, then the resultant DBCS value in the DB2/400 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 DB2/400 column by "wā" 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 Data Area (or Environment variable) controls this feature. If there is no ORADBMBOPT Data Area (or Environment variable), then no forcing of single-byte character to double-byte character will take place when transferring data into DB2/400 columns, and no forcing of double-byte characters to the single-byte correspondences will take place when transferring data from DB2/400 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 DB2/400 graphic constants. DB2/400 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.

**Performing Character String Operations**

The gateway performs all character string comparisons, concatenations, and sorts using the datatype of the referenced columns (for example, fixed or variable length). The gateway determines the validity of character string values passed by applications and automatically converts character strings from one datatype to another. It also automatically converts between character strings and dates when necessary.

In contrast, the Oracle9i database server processes character strings as variable length character strings. The Oracle server performs character string comparisons, concatenations, and sorts using variable length character string representations. The Oracle server never pads or truncates character strings. The Oracle server also automatically converts among numbers, character strings, and dates when necessary.
Converting Character String Datatypes

The gateway binds character string data values from host variables as fixed length character strings. The bind length is the length of the character string data value. This conversion is performed on every bind.

DB2/400 performs automatic conversions to the character string datatype (for example, fixed length and variable length) of the destination column. You have no control over the conversion. The conversion might be independent of the datatype of the destination column in the database. For example:

```
UPDATE EMP@AS400 SET ENAME = 'BLAKE' WHERE EMPNO = 7654;
```

If ENAME is a CHAR(9), fixed-length column in the ENAME table, then the update inserts 'BLAKE' as the last name of employee number 7654. BLAKE is padded with four trailing spaces by the host database to make it conform to the fixed length definition of the column.

The gateway returns data bound in the manner requested by the application. For character strings, the format is fixed length string.

The DB2/400 VARCHAR datatype can be from 1 to 32740 bytes in length. This datatype is converted to an Oracle VARCHAR2 datatype if it is between 1 and 4000 characters in length. If it is between 4001 and 32740 characters in length, it is converted to an Oracle LONG datatype.

The Oracle LONG datatype can be from 1 byte to 2 GB in length. Because the DB2/400 VARCHAR datatype can be no longer than 32740 bytes, you receive an error message if you attempt to insert data longer than 32740 bytes into a DB2/400 VARCHAR or LONG VARCHAR datatype.

Performing Date and Time Operations

The implementation of date and time data differs significantly in DB2/400 and the Oracle server. The Oracle9i database server has a single date datatype, DATE, that can contain both calendar date and time of day information. DB2/400 supports the following date and time datatypes:

```
DATE is the calendar date only.
TIME is the time of day only.
TIMESTAMP is a numerical value combining a calendar date and time of day with microsecond resolution.
```
There is no mechanism that translates the IBM TIME and TIMESTAMP data to Oracle DATE data. An application must process TIME datatypes in the Oracle CHAR format with a length of 8 bytes. An application must process the TIMESTAMP datatype in the Oracle CHAR format with a length of 26 bytes.

An application reads TIME and TIMESTAMP functions as character strings and converts portions of the string to perform numerical operations. TIME and TIMESTAMP values can be sent to DB2/400 as character literals or bind variables of the appropriate length and format.

Oracle datatypes and IBM DATE datatypes are mapped to each other. If an IBM DATE is queried, 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, the date value is converted to the IBM DATE format and any time value is discarded. Columns of DATE or TIME datatype are in ISO format. For DATE, the column format is "yyyy-mm-dd", and for TIME columns, the format is "hh.mm.ss". TIME is in a 24-hour format.

Character representations of dates are different in Oracle format and DB2/400 format. When an Oracle application SQL statement contains a date literal or conveys a date through a character bind variable, the gateway must convert the date to a DB2/400 compatible format.

The gateway does not automatically recognize when a character value is going to be processed against an IBM DATE column. Applications are required to distinguish character date values by enclosing them with Oracle TO_DATE( ) function notation.

For example, if EMP is a synonym or view accessing data in DB2/400, instead of the following SQL statement:

SELECT * FROM EMP@AS400 WHERE HIREDATE = '03-MAR-81'

you must use:

SELECT * FROM EMP@AS400 WHERE HIREDATE = TO_DATE('03-MAR-81')

In a programmatic interface program using a character bind variable for the qualifying date value, you must use the following SQL statement:

SELECT * FROM EMP@AS400 WHERE HIREDATE = TO_DATE(:1)

This SQL notation does not affect SQL statement semantics when the statement is executed against an Oracle table. The statement remains the same across Oracle and DB2/400 data stores.
The DATE requirement does not apply to input bind variables in Oracle date seven byte binary format. The gateway recognizes such values to be dates.

**Note:** There are special considerations for using date or time datatypes in stored procedures with DB2/400. Refer to "Known Problems" on page 2-4 for more information.

### Dates in the 21st Century

Oracle Corporation recommends that you set the Oracle9i database server default NLS_DATE_FORMAT parameter to a format including a four-digit year. Use the TO_DATE function to enter dates in the twenty-first century. You can use any date format which includes a four-character field for the year. For example, TO_DATE('2008-07-23', 'YYYY-MM-DD') can be used in any SELECT, INSERT, UPDATE, or DELETE statement.

### Performing Numeric Datatype Operations

DB2/400 performs automatic conversions to the numeric datatype of the destination column such as INTEGER, SMALLINT, PACKED DECIMAL, and NUMERIC. You have no control over the datatype conversion, and the conversion is independent of the datatype of the destination column in the database.

For example, if PRICE is an integer column of the PRODUCT table in DB2/400, the update shown in this example inaccurately sets the price of an ice cream cone to $1.00, because DB2/400 automatically converts a double-precision floating point to an integer:

```
UPDATE PRODUCT@AS400
SET PRICE = 1.50
WHERE PRODUCT_NAME = 'ICE CREAM CONE';
```

Because PRICE is an integer, the AS/400 server automatically converts the decimal data value of 1.50 to 1.

The gateway returns values bound in the manner requested by the Oracle application. For numbers, the format is typically an Oracle number, limited to a certain number of digits of precision by the application.
One of the most important features of the Oracle Enterprise Gateway product family is the ability to provide SQL transparency to the end user and to the application programmer. Foreign data store SQL functions can be categorized into three areas:

1. Compatible

Compatible SQL functions have the same meaning and results on both Oracle and foreign data store. Compatible SQL functions include both column and scalar functions.

   Column Functions
   - AVG
   - COUNT (*) only
   - COUNT (DISTINCT expression)
   - MAX
   - MIN
   - STDDEV
   - SUM

   Scalar Functions
   - ABS
   - COS
   - COSH
   - EXP
   - LENGTH (char) only
   - LN
   - MOD
   - POWER
   - SIN
   - SINH
   - SQRT
SQL Functions

- TAN
- TANH
- UPPER

2. Translated

Translated SQL functions provide the same functionality, but are referenced by different names at the Oracle server and the foreign data store.

Translated SQL functions include:

<table>
<thead>
<tr>
<th>Oracle</th>
<th>DB2/400</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVL</td>
<td>VALUE</td>
</tr>
<tr>
<td>LOWER</td>
<td>TRANSLATE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Use the passthrough feature for native DB2/400 SQL statements when updating the VALUE function. Refer to "Passing DB2/400 SQL Statements through the Gateway" on page 9-11 for additional information.

3. Compensated

Compensated SQL functions are advanced SQL functions that are supported by the Oracle database and that cannot be expressed or recognized by the foreign data store.

SQL compensation in the Oracle Enterprise Gateways enriches the semantics of the native SQL of a remote data source, such as DB2/400. This important feature of the gateway allows application developers and end-users to leverage the advanced features of the Oracle database.

Oracle SQL functions that are not listed in the compatible or translated lists above are compensated SQL functions.
Oracle Server SQL Construct Processing

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 DB2/400, it is given to DB2/400 to perform.

■ If part, but not all, of the WHERE clause of the SELECT statement uses features not available in DB2/400, the WHERE clause is split between the DB2/400 system and the Oracle server.

   The portion of the WHERE clause acceptable for DB2/400 is sent to DB2/400. The Oracle server post-processes the results of the DB2/400 SELECT and applies the Oracle-specific WHERE clause elements. This results in DB2/400 doing as much of the WHERE clause as possible.

■ If the entire WHERE clause is not acceptable for DB2/400, an unqualified SELECT (without the WHERE clause) is sent for DB2/400, and the Oracle server post-processes the entire WHERE clause.

The Oracle 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

DB2/400 must process the entire SELECT FOR UPDATE, INSERT, and DELETE clauses. The Oracle server cannot post-process these clauses. Only SQL that is a common subset of Oracle and DB2/400 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 DB2/400 can be used. Refer to the IBM reference for DB2/400 for DB2/400 SQL syntax.
- The following Oracle 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 DB2/400 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 Server and DB2/400 Differences

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, DB2/400 sets the field to -1 (negative 1) even if more rows are actually deleted.

Oracle Bind Variables

Oracle bind variables become DB2/400 parameter markers when used with the gateway. Therefore, the bind variables are subject to the same restrictions as DB2/400 parameter markers. For example, the following statement fragments are not allowed:

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

Refer to the IBM reference for DB2/400 for more information about DB2/400 parameter marker restrictions.

Mapping the COUNT Function

The Oracle database server supports four options for the COUNT function:

- `COUNT(*)`
- `COUNT(DISTINCT expression)`
- `COUNT(ALL expression)`
- `COUNT(expression)`

`COUNT(ALL expression)` and `COUNT(expression)` are post-processed. DB2/400 servers support only two options for the COUNT function:

- `COUNT(*)`
- `COUNT(DISTINCT expression)`
Performing Zoned Decimal Operations

A zoned decimal field is described as packed decimal on an Oracle server. However, an Oracle application such as a Pro*C program can insert any supported Oracle numeric datatype into a zoned decimal column. The gateway converts this number into the most suitable datatype. Data can be retrieved from the AS/400 into any Oracle datatype provided it does not result in a loss of information.

Oracle Data Dictionary Emulation in a DB2/400 Server

The gateway can optionally augment DB2/400 database catalogs with data dictionary views modeled after the Oracle data dictionary. These views are based upon the dictionary tables in DB2/400, 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 you with an Oracle-like interface to the contents and use of DB2/400. Some of these views are required by Oracle products.

You can query the gateway data dictionary views to see the objects in DB2/400 and to determine the authorized users of those objects.

Refer to Appendix C, "Data Dictionary Views" for descriptions of DB2/400 catalog views.

DB2/400 Special Registers

You can access DB2/400 special registers using the gateway. During installation of the gateway, a DB2/400 view is created to access special registers. For example, in order to find out the primary authorization ID being used by the gateway, execute this command from your application:

```
SELECT CURRENT_USER FROM OTGDB2.OTGREGISTER@DB2400
```

where OTGDB2 is the default qualifier of the OTGREGISTER view, and DB2400 is the name of a database link to the gateway.
Oracle Developer Forms Compatibility

You can use Oracle Developer Forms version 4 or higher to build applications stored in DB2/400. When you connect to a non-Oracle data source with a gateway product, you have four transaction processing options in Oracle Developer Forms of which you should be aware. These options include two block properties and two form module properties. When designing the form, set these options as described below:

<table>
<thead>
<tr>
<th>Option</th>
<th>Set Value To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Mode block property</td>
<td>Anything except Unique_key</td>
</tr>
<tr>
<td>Locking Mode block property</td>
<td>Delayed</td>
</tr>
<tr>
<td>Cursor Mode form module</td>
<td>Close_at_commit</td>
</tr>
<tr>
<td>Savepoint Mode form module</td>
<td>Off</td>
</tr>
</tbody>
</table>

You should also ensure that at item and block level the primary key is set to TRUE. For more information refer to your Oracle Forms documentation.
This chapter discusses error messages generated by Oracle Transparent Gateway for DB2/400, the diagnosis of suspected Oracle errors, and the requirements for documenting these errors to Oracle Support Services.

This chapter contains the following sections:

- Message and Error Code Processing on page 10-2
- Contacting Oracle Support Services on page 10-4
- Error Categories on page 10-5
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 that refers to one or more DB2/400 database tables. An error condition can be complex, involving error codes and supporting data from multiple components. In all cases, the application receives a single Oracle error code upon which to act.

When possible, an error code from DB2/400 is converted to the Oracle 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 that are common to most relational databases.

Mapping DB2/400 Error Messages to Oracle Error Messages

DB2/400 error messages, that is, SQLSTATE codes, are mapped to Oracle9i database server error codes. Notice that multiple DB2/400 SQLSTATE codes can refer to the same single Oracle database server error code.

<table>
<thead>
<tr>
<th>Description</th>
<th>SQLSTATE Code</th>
<th>Oracle Database Server Error Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>No rows selected</td>
<td>0200</td>
<td>0</td>
</tr>
<tr>
<td>Unique index constraint violated</td>
<td>23505</td>
<td>ORA-0001</td>
</tr>
<tr>
<td>Table or view does not exist</td>
<td>52004 or 42704</td>
<td>ORA-00942</td>
</tr>
<tr>
<td>Object name greater than 18 characters and, therefore, object does not exist</td>
<td>54003 or 42622</td>
<td>ORA-00942</td>
</tr>
<tr>
<td>Insufficient privileges</td>
<td>42501</td>
<td>ORA-01031</td>
</tr>
<tr>
<td>Divisor is equal to zero</td>
<td>01519 or 01564</td>
<td>ORA-01476</td>
</tr>
</tbody>
</table>

Interpreting Gateway Message Formats

Error messages are generally accompanied by additional message text, beyond the text associated with the Oracle 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 OCIErrorGet in the programmatic and Oracle Call Interfaces you use with the gateway to view the entire message. Refer to the PL/SQL User’s Guide and Reference for information about SQLGLM and the Oracle Call Interface Programmer’s Guide for information about OCIErrorGet.

Gateway messages use this format:

ORA-nnnn error_message_text
DB2/400_error_messages
ORA-2063: preceding n lines from dblink

for example:

select * from scott.dummy@DB2link
ERROR at line 1:
ORA-00942 = table or view does not exist
SQL0204: DUMMY in SCOTT type * FILE not found
ORA-02063 preceding 2 lines from dblink

where:

nnnn is an Oracle error number. If nnnn is between 28500 and 28559, the message is from the gateway. If it is not in this range, it is a mapped error message. In the example above, the value for nnnn is 00942

error_message_text is the text of the message associated with the error. In the example above the value for error_message_text is table or view does not exist.

DB2/400_error_messages are additional messages generated by DB2/400. In the example above, the value for DB2/400_error_messages is SQL0204 : DUMMY in SCOTT type * FILE not found

n is the total number of DB2/400 error_messages lines referenced above. In the example above the value for n is 2.

dblink is the name of the database link used to access the gateway. In the example above the value for dblink is DB2link.
Diagnosing Errors Detected by the Oracle9i database server

If an error is detected by the Oracle9i database server, the DB2/400 error messages do not occur. For example, if the gateway cannot be accessed because of an Oracle Net or gateway installation problem, the DB2/400 error message is not present in the error message.

Another example of error messages without DB2/400 error messages 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. This SQL statement causes an error message:

```
SQL> INSERT INTO EMP@AS400 VALUES(9999);
ERROR at line 1:
ORA-00947: NOT ENOUGH VALUES
```

The ORA-00947 message is not accompanied by DB2/400 error messages because the error is detected by the Oracle9i database server. The server obtains a description of the DB2/400 table before sending the INSERT statement to the gateway for processing. This allows the server to detect when the INSERT statement is invalid.

Contacting Oracle Support Services

To maximize the effectiveness of your inquiries, gather the following information before contacting Oracle Support Services:

- OS/400 release number
- gateway release number (found in the JobLog for a Gateway task)
- Oracle Net release number (found in the JobLog for a Gateway task)
- Oracle database server release number and platform
Error Categories

Use the following error categories to describe the error:

- documentation errors
- incorrect output
- Oracle external error
- abnormal termination
- 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, 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.)
- What are the exact SQL statements used to reproduce the problem?
- What are the full version numbers of the AS/400 system Oracle database, Oracle gateway, and related Oracle software?
- What is the problem and how is it reproduced?

Oracle External Error

Oracle error messages are produced whenever an Oracle gateway, server, tool, or DB2/400 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.
Performance

AS/400 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 DB2/400 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 workload characterization
- query execution plans
- DB2/400 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.
Oracle Net Files and Members

This appendix lists and defines the member names in the LOG, ORA, and TRC gateway library files.

This appendix contains the following sections:

- LOG File Members on page A-2
- ORA File Members on page A-2
- TRC File Members on page A-3
- Agent Trace Files on page A-3
LOG File Members

The LOG file contains the LISTENER member, which is the default log output member for the Oracle Net TCP/IP listener.

To display this member on the AS/400, use the following command:

```
WRKMBRPDM instance_name/LOG
```

where `instance_name` is the name used for the gateway when it is installed.

ORA File Members

The ORA file contains the following members:

- **LISTENER** contains the parameter values for the listener. The values in this file cannot be changed directly with an AS/400 editor. The most commonly used parameters can be changed with the CHGORANET command. All other values must be changed in the ORA(LISTENER_CHG) file. Refer to "CHGORANET, Change Network Parameters" on page 6-8 for more information.

- **LISTENER_CHG** is used to make changes to the ORA(LISTENER) file. Refer to "CHGORANET, Change Network Parameters" on page 6-8 for more information.

- **LISTENLOCK** is the ORA(LISTENLOCK) member. It is used to ensure that only one instance of the RUNORALS program is running. The RUNORALS program runs the TCP/IP listener. Do not modify this file.

- **SQLNET** contains the network parameter values for Oracle Net. The values in this file cannot be changed directly with an AS/400 editor. The most commonly used parameters can be changed with the CHGORANET command. All other values must be changed in the ORA(SQLNET_CHG) file. Refer to "CHGORANET, Change Network Parameters" on page 6-8 for more information.

- **SQLNET_CHG** is used to make changes to the ORA(SQLNET) file. Refer to "CHGORANET, Change Network Parameters" on page 6-8 for more information.
To display these members on the AS/400, use this command:

`WRKMBRPDM instance_name/ORA`

where `instance_name` is the name used for the gateway when it is installed. You may also use the command WRKTRCLOG, but you must first make sure that the instance library is in the library list before using this command.

**TRC File Members**

The TRC file contains the following members, where `xxxxxx` is numeric (the job number):

- `LISTxxxxxx` is the default trace output file for the Oracle Net TCP/IP listener, where `xxxxxx` is the job number of the listener.
- `SERVxxxxxx` is the Oracle Net trace for server job number `xxxxxx`.

To display these members on the AS/400, use the following command:

`WRKMBRPDM instance_name/TRC`

where `instance_name` is the name used for the gateway when it is installed. Or you may use the command WRKTRCLOG, but you must first make sure that the instance library is in the library list before using this command.

**Agent Trace Files**

Agent trace files are now located within the integrated file system in the directory `/home/xxxxxx/rdbms/log`, where `xxxxxx` is the instance name. The filenames are of the form `xxxxxx_agt_nnnn.nnnnn.trc`, where `xxxxxx` is the instance name and `nnnnnn` is the job number. To display these files, use the WRKTRCLOG command, but ensure that the instance library is in the library list before doing so.
This appendix documents the National Language Support (NLS) information for the Oracle Transparent Gateway for DB2/400. For more information about using NLS refer to *Oracle9i Application Developer’s Guide - Fundamentals*.

This appendix contains the following sections:

- Overview of National Language Support on page B-2
- Gateway Configuration on page B-2
- DB2/400 Coded Character Set Considerations on page B-3
- Oracle Server and Client Configuration on page B-6
- Message Availability on page B-6
- DB2/400 GRAPHIC Support on page B-6
Overview of 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 Oracle NLS architecture is data-driven, enabling support for specific languages and character encoding schemes to be added without requiring any changes in source code.

There are a number of different settings in the gateway, DB2/400, Oracle server, and the client that affect NLS processing. In order for translations to take place correctly, character settings of these components must be compatible. Each character in one encoding scheme must have a matching character in another encoding scheme.

Gateway Configuration

Once the gateway is installed, you must use the CHGORATUN command if you need to change language settings.

CHGORATUN Language Specification

The CHGORATUN command allows you to change the language parameter that defines the character set used for the gateway. The language parameter entered with this command specifies the conventions such as language used for gateway messages, names of days and months, symbols for AD, BC, AM, and PM, and the default language sorting mechanism.

The syntax for specifying the language parameter is:

```
language[_territory.character_set]
```

- **language** is any valid language documented in “Supported Languages and Territories” on page B-5.
- **territory** is optional and defaults to AMERICA. Valid values are documented in “Supported Languages and Territories” on page B-5.
- **character_set** is optional and defaults to WE8EBCDIC37. Valid values are documented in “Supported Character Sets” on page B-3.

The default setting is:

```
AMERICAN_AMERICA.WE8EBCDIC37
```

To change this setting, use the Gateway language field on the CHGORATUN main menu panel. The gateway must be shutdown and restarted before the new settings take effect.

B-2  Oracle Transparent Gateway for DB2/400 Installation and User’s Guide
parameter takes effect. For more information about the CHGORATUN command, refer to "CHGORATUN, Change Initialization Parameters" on page 6-10.

**DB2/400 Coded Character Set Considerations**

If the coded character set identifier (CCSID) of the AS/400 data file is different from 65535, then the Oracle language parameters must match the CCSID of the AS/400 data file that is being accessed. For example, if the CCSID is 280 for Italy, then the Oracle character set must be set to I8EBCDIC280. The exception to this is for columns with a CCSID of 13488 (UCS-2). the data in these columns will always be converted to the character set that is determined by NLS_LANG. Contact your DBA or refer to the IBM manual for AS/400 National Language Support for additional information about AS/400 CCSID codes.

**Supported Character Sets**

Oracle Transparent Gateway for DB2/400 supports the following languages and values for `character_set`.

Note that the character sets marked with an asterisk (*) are the Euro versions of the immediately preceding character set.

<table>
<thead>
<tr>
<th>Description</th>
<th>Character Set</th>
<th>OS/400 CCSID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austrian/German</td>
<td>D8EBCDIC273</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>D8EBCDIC1141*</td>
<td>1141</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>ZHT16DBCS</td>
<td>937</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>ZHS16DBCS</td>
<td>935</td>
</tr>
<tr>
<td>Danish/Norwegian</td>
<td>DK8EBCDIC277</td>
<td>277</td>
</tr>
<tr>
<td></td>
<td>DK8E8CDIC1142*</td>
<td>1142</td>
</tr>
<tr>
<td>Eastern European</td>
<td>EE8EBCDIC870</td>
<td>870</td>
</tr>
<tr>
<td>Finnish/Swedish</td>
<td>S8EBCDIC278</td>
<td>278</td>
</tr>
<tr>
<td></td>
<td>S8EBCDIC1143*</td>
<td>1143</td>
</tr>
<tr>
<td>French/France</td>
<td>F8EBCDIC297</td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>F8EBCDIC1147*</td>
<td>1147</td>
</tr>
<tr>
<td>German/German</td>
<td>D8EBCDIC273</td>
<td>273</td>
</tr>
<tr>
<td>Description</td>
<td>Character Set</td>
<td>OS/400 CCSID</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Greek</td>
<td>EL8EBCDIC875</td>
<td>875</td>
</tr>
<tr>
<td>Hebrew</td>
<td>IW8EBCDIC424</td>
<td>424</td>
</tr>
<tr>
<td>Italian</td>
<td>I8EBCDIC280</td>
<td>280</td>
</tr>
<tr>
<td></td>
<td>I8EBCDIC1144*</td>
<td>1144</td>
</tr>
<tr>
<td>Japanese</td>
<td>JA16DBCS</td>
<td>939, 5035</td>
</tr>
<tr>
<td></td>
<td>JA16EBCDIC930</td>
<td>930, 5026</td>
</tr>
<tr>
<td>Korean</td>
<td>KO16DBCS</td>
<td>933</td>
</tr>
<tr>
<td>Spanish</td>
<td>WE8EBCDIC284</td>
<td>284</td>
</tr>
<tr>
<td>Thai</td>
<td>TH8TISEBCDIC</td>
<td>838</td>
</tr>
<tr>
<td>Turkish</td>
<td>TR8EBCDIC1026</td>
<td>1026</td>
</tr>
<tr>
<td>Western European</td>
<td>WE8EBCDIC37</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC1140*</td>
<td>1140</td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC37C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC500</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC1148*</td>
<td>1148</td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC500C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC1148C*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC285</td>
<td>285</td>
</tr>
<tr>
<td></td>
<td>WE8EBCDIC1146*</td>
<td>1146</td>
</tr>
</tbody>
</table>
## Supported Languages and Territories

Oracle Transparent Gateway for DB2/400 supports these language and territory combinations:

<table>
<thead>
<tr>
<th>Language</th>
<th>Territory</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>America</td>
</tr>
<tr>
<td>Brazilian Portuguese</td>
<td>Brazil</td>
</tr>
<tr>
<td>Canadian French</td>
<td>Canada</td>
</tr>
<tr>
<td>Czech</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>Danish</td>
<td>Denmark</td>
</tr>
<tr>
<td>Dutch</td>
<td>Netherlands</td>
</tr>
<tr>
<td>Finnish</td>
<td>Finland</td>
</tr>
<tr>
<td>French</td>
<td>France</td>
</tr>
<tr>
<td>German</td>
<td>Germany</td>
</tr>
<tr>
<td>Greek</td>
<td>Greece</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Hungary</td>
</tr>
<tr>
<td>Icelandic</td>
<td>Iceland</td>
</tr>
<tr>
<td>Italian</td>
<td>Italy</td>
</tr>
<tr>
<td>Japanese</td>
<td>Japan</td>
</tr>
<tr>
<td>Mexican Spanish</td>
<td>Mexico</td>
</tr>
<tr>
<td>Norwegian</td>
<td>Norway</td>
</tr>
<tr>
<td>Polish</td>
<td>Poland</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Portugal</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>China</td>
</tr>
<tr>
<td>Slovak</td>
<td>Slovakia</td>
</tr>
<tr>
<td>Spanish</td>
<td>Spain</td>
</tr>
<tr>
<td>Swedish</td>
<td>Sweden</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>Taiwan</td>
</tr>
<tr>
<td>Turkish</td>
<td>Turkey</td>
</tr>
</tbody>
</table>
Oracle Server and Client Configuration

There are a number of NLS parameters that control NLS processing between the Oracle server and the 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 the Oracle9i SQL Reference. These parameters do not affect gateway processing. However, you must ensure the character set is compatible with the character set you specify on the gateway and DB2/400. 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 recreate the database.

Normally US7ASCII is the default for CHARACTER SET on non-EBCDIC platforms. US7ASCII only supports 26 Latin alphabetic characters. If you have specified 8-bit character sets on the gateway and DB2/400, you must have a compatible 8-bit character set defined for your database. To check the character set of an existing database, issue this command in SQL*Plus:

```
SELECT userenv('language') FROM DUAL
```

Message Availability

Availability of the supported language message modules depends on which modules are installed in the Oracle product set that is running on the server. If you do not have message modules installed for a particular language set, then specifying that language with a language parameter results in no messages being displayed for that module in the requested language. Only a generalized (and rather uninformative) message will be provided.

DB2/400 GRAPHIC Support

When converting DB2/400 datatypes to Oracle datatypes, if support for DB2/400 GRAPHIC datatypes (GRAPHIC, VARGRAPHIC, or LONG VARGRAPHIC) is required, special consideration must be given to the selection of the NLS_LANG character set. Refer to "DB2/400 GRAPHIC Support" on page 9-17 for more information.
This appendix includes the gateway data dictionary views accessible to all users of an Oracle server. Most views can be accessed by any user with SELECT privileges for DB2/400 catalog tables.

A dictionary item described with N/A in this appendix means that the item is not available for the gateway. Such items default to zero for numeric columns, or to blanks or some fixed value for (VAR)CHAR columns.

This appendix contains the following sections:

- **ALL_CATALOG** on page C-3
- **ALL_COL_COMMENTS** on page C-3
- **ALL_CONSTRAINTS** on page C-4
- **ALL_IND_COLUMNS** on page C-5
- **ALL_INDEXES** on page C-5
- **ALL_OBJECTS** on page C-7
- **ALL_SYNONYMS** on page C-7
- **ALL_TAB_COLUMNS** on page C-8
- **ALL_TAB_COMMENTS** on page C-9
- **ALL_TABLES** on page C-9
- **ALL_USERS** on page C-10
- **ALL_VIEWS** on page C-11
- **COLUMN_PRIVILEGES** on page C-11
- **OTGREGISTER** on page C-12
■ TABLE_PRIVILEGES on page C-13
■ USER_CATALOG on page C-13
■ USER_COL_COMMENTS on page C-13
■ USER_CONS_COLUMNS on page C-14
■ USER_CONSTRAINTS on page C-14
■ USER_INDEXES on page C-15
■ USER_OBJECTS on page C-16
■ USER_SYNONYMS on page C-17
■ USER_TAB_COLUMNS on page C-18
■ USER_TAB_COMMENTS on page C-19
■ USER_TABLES on page C-19
■ USER_USERS on page C-21
■ USER_VIEWS on page C-21
The column names listed for the following views are the column names that the Oracle Database server presents to the client and may actually differ from the underlying views in DB2/400.

**ALL_CATALOG**

All tables, views, synonyms, and sequences accessible to the user.

<table>
<thead>
<tr>
<th>OWNER</th>
<th>Schema of the object</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of object</td>
</tr>
</tbody>
</table>

**ALL_COL_COMMENTS**

Comments on columns of accessible tables and views.

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

Constraint definitions on accessible tables.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>Name associated with the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_TYPE</td>
<td>Type of constraint definition</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name associated with the table with the constraint definition</td>
</tr>
<tr>
<td>SEARCH_CONDITION</td>
<td>Text of the search condition for the table check</td>
</tr>
<tr>
<td>R_OWNER</td>
<td>Owner of the table used in the referential constraint</td>
</tr>
<tr>
<td>R_CONSTRAINT_NAME</td>
<td>Name of the unique constraint definition for the referenced table</td>
</tr>
<tr>
<td>DELETE_RULE</td>
<td>Delete rule for the referential constraint</td>
</tr>
<tr>
<td>STATUS</td>
<td>Status of the constraint</td>
</tr>
<tr>
<td>DEFERRABLE</td>
<td>N/A</td>
</tr>
<tr>
<td>DEFERRED</td>
<td>N/A</td>
</tr>
<tr>
<td>VALIDATED</td>
<td>N/A</td>
</tr>
<tr>
<td>GENERATED</td>
<td>N/A</td>
</tr>
<tr>
<td>BAD</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_CHANGE</td>
<td>N/A</td>
</tr>
</tbody>
</table>
ALL_IND_COLUMNS

Columns of the indexes on the accessible tables.

<table>
<thead>
<tr>
<th>INDEX_OWNER</th>
<th>Owner of the index</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX_NAME</td>
<td>Name of the index</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Table owner</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Table name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>COLUMN_POSITION</td>
<td>Position of column within the index</td>
</tr>
<tr>
<td>COLUMN_LENGTH</td>
<td>Indexed length of column</td>
</tr>
</tbody>
</table>

ALL_INDEXES

Description of indexes on tables accessible to the user.

<table>
<thead>
<tr>
<th>OWNER</th>
<th>Owner of the index</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX_NAME</td>
<td>Name of the index</td>
</tr>
<tr>
<td>INDEX_TYPE</td>
<td>Type of the Index</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the indexed object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the indexed object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of indexed object</td>
</tr>
<tr>
<td>UNIQUENESS</td>
<td>Uniqueness status of the index</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the index</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### ALL_INDEXES

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCT_THRESHOLD</td>
<td>N/A</td>
</tr>
<tr>
<td>INCLUDE_COLUMN</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING</td>
<td>N/A</td>
</tr>
<tr>
<td>BLEVEL</td>
<td>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.</td>
</tr>
<tr>
<td>LEAF_BLOCKS</td>
<td>Number of leaf blocks in the index</td>
</tr>
<tr>
<td>DISTINCT_KEYS</td>
<td>Number of distinct indexed values. For indexes enforcing UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table.</td>
</tr>
<tr>
<td>AVG_LEAF_BLOCKS_PER_KEY</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_DATA_BLOCKS_PER_KEY</td>
<td>N/A</td>
</tr>
<tr>
<td>CLUSTERING_FACTOR</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the index: VALID</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>N/A</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>N/A</td>
</tr>
<tr>
<td>DEGREE</td>
<td>N/A</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>N/A</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>N/A</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>N/A</td>
</tr>
<tr>
<td>GENERATED</td>
<td>N/A</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### ALL_OBJECTS

Objects accessible to the user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the object</td>
</tr>
<tr>
<td>OBJECT_NAME</td>
<td>Name of object</td>
</tr>
<tr>
<td>SUBOBJECT_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>OBJECT_ID</td>
<td>Object number of the object</td>
</tr>
<tr>
<td>OBJECT_TYPE</td>
<td>Type of object</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_DDL_TIME</td>
<td>N/A</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the object</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>N/A</td>
</tr>
<tr>
<td>GENERATED</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### ALL_SYNONYMS

All synonyms accessible to the user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the synonym</td>
</tr>
<tr>
<td>SYNONYM_NAME</td>
<td>Name of the synonym</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the object referenced by the synonym</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object referenced by the synonym</td>
</tr>
<tr>
<td>DB_LINK</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Columns of all tables, views, and clusters accessible to the user.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the table or view</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Table or view name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>Datatype of the column</td>
</tr>
<tr>
<td>DATA_TYPE_MOD</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_TYPE_OWNER</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_LENGTH</td>
<td>Maximum length of the column in bytes</td>
</tr>
<tr>
<td>DATA_PRECISION</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_SCALE</td>
<td>Digits to the right of decimal point in a number</td>
</tr>
<tr>
<td>NULLABLE</td>
<td>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.</td>
</tr>
<tr>
<td>COLUMN_ID</td>
<td>Sequence number of the column as created</td>
</tr>
<tr>
<td>DEFAULT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_DEFAULT</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_DISTINCT</td>
<td>Number of distinct values in each column of the table</td>
</tr>
<tr>
<td>LOW_VALUE</td>
<td>Second lowest and the 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.</td>
</tr>
<tr>
<td>HIGH_VALUE</td>
<td></td>
</tr>
<tr>
<td>DENSITY</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_NULLS</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_BUCKETS</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>N/A</td>
</tr>
<tr>
<td>SAMPL_SIZE</td>
<td>N/A</td>
</tr>
<tr>
<td>CHARACTER_SET_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>CHAR_COL_DECL_LENGTH</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### ALL_TAB_COMMENTS

Comments on tables and views accessible to the user.

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

### ALL_TABLES

Description of tables accessible to the user.

<table>
<thead>
<tr>
<th>OWNER</th>
<th>Owner of the table</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Name of the table</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the table</td>
</tr>
<tr>
<td>CLUSTER_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>IOT_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_USED</td>
<td>N/A</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING</td>
<td>N/A</td>
</tr>
<tr>
<td>BACKED_UP</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### ALL_USERS

Information about all users of the database.

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>USER_ID</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### COLUMN_PRIVILEGES

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

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

### ALL_VIEWS

Text of views accessible to the user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the view</td>
</tr>
<tr>
<td>VIEW_NAME</td>
<td>Name of the view</td>
</tr>
<tr>
<td>TEXT_LENGTH</td>
<td>Length of the view text</td>
</tr>
<tr>
<td>TEXT</td>
<td>View text. Only the first row of text is returned, even if multiple rows exist.</td>
</tr>
<tr>
<td>TYPE_TEXT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>TYPE_TEXT</td>
<td>N/A</td>
</tr>
<tr>
<td>OID_TEXT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>OID_TEXT</td>
<td>N/A</td>
</tr>
<tr>
<td>VIEW_TYPE_OWNER</td>
<td>N/A</td>
</tr>
<tr>
<td>VIEW_TYPE</td>
<td>N/A</td>
</tr>
</tbody>
</table>
DB2/400 Special Registers

<table>
<thead>
<tr>
<th>Special Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT_USER</td>
<td>Primary authorization id</td>
</tr>
<tr>
<td>CURRENT_DATE</td>
<td>Current date</td>
</tr>
<tr>
<td>CURRENT_TIME</td>
<td>Current time</td>
</tr>
<tr>
<td>CURRENT_TIMESTAMP</td>
<td>Current timestamp</td>
</tr>
<tr>
<td>CURRENT_TIMEZONE</td>
<td>Current timezone</td>
</tr>
<tr>
<td>CURRENT_SERVER</td>
<td>Host name of the current server</td>
</tr>
<tr>
<td>&quot;current_user&quot;</td>
<td>Primary authorization id</td>
</tr>
<tr>
<td>&quot;current_date&quot;</td>
<td>Current date</td>
</tr>
<tr>
<td>&quot;current_time&quot;</td>
<td>Current time</td>
</tr>
<tr>
<td>&quot;current_timestamp&quot;</td>
<td>Current timestamp</td>
</tr>
<tr>
<td>&quot;current_timezone&quot;</td>
<td>Current timezone</td>
</tr>
<tr>
<td>&quot;current_server&quot;</td>
<td>Host name of the current server</td>
</tr>
</tbody>
</table>
**TABLE_PRIVILEGES**

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

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

**USER_CATALOG**

Tables, views, synonyms, and sequences owned by the user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of object</td>
</tr>
</tbody>
</table>

**USER_COL_COMMENTS**

Comments on columns of user's tables and views.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Object name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Comments on the column</td>
</tr>
</tbody>
</table>
## USER_CONS_COLUMNS

Information about columns in constraint definitions owned by the user.

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

## USER_CONSTRAINTS

Constraint definitions on user’s tables.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>Owner of the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>Name associated with the constraint definition</td>
</tr>
<tr>
<td>CONSTRAINT_TYPE</td>
<td>Type of constraint definition</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name associated with the table with the constraint definition</td>
</tr>
<tr>
<td>SEARCH_CONDITION</td>
<td>Text of the search condition for the table check</td>
</tr>
<tr>
<td>R_OWNER</td>
<td>Owner of the table used in the referential constraint</td>
</tr>
<tr>
<td>R_CONSTRAINT_NAME</td>
<td>Name of the unique constraint definition for the referenced table</td>
</tr>
<tr>
<td>DELETE_RULE</td>
<td>Delete rule for the referential constraint</td>
</tr>
<tr>
<td>STATUS</td>
<td>Status of the constraint</td>
</tr>
<tr>
<td>DEFERRABLE</td>
<td>N/A</td>
</tr>
<tr>
<td>DEFERRED</td>
<td>N/A</td>
</tr>
<tr>
<td>VALIDATED</td>
<td>N/A</td>
</tr>
<tr>
<td>GENERATED</td>
<td>N/A</td>
</tr>
<tr>
<td>BAD</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_CHANGE</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### USER_INDEXES

Description of the user's own indexes.

<table>
<thead>
<tr>
<th>INDEX_NAME</th>
<th>Name of the index</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX_TYPE</td>
<td>Type of Index</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the indexed object</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the indexed object</td>
</tr>
<tr>
<td>TABLE_TYPE</td>
<td>Type of the indexed object</td>
</tr>
<tr>
<td>UNIQUENESS</td>
<td>Uniqueness status of the index</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the index</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_THRESHOLD</td>
<td>N/A</td>
</tr>
<tr>
<td>INCLUDE_COLUMN</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING</td>
<td>N/A</td>
</tr>
<tr>
<td>BLEVEL</td>
<td>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.</td>
</tr>
<tr>
<td>LEAF_BLOCKS</td>
<td>Number of leaf blocks in the index</td>
</tr>
<tr>
<td>DISTINCT_KEYS</td>
<td>Number of distinct indexed values. For indexes enforcing UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table.</td>
</tr>
</tbody>
</table>
### USER_OBJECTS

Objects owned by the user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG_LEAF_BLOCKS_PER_KEY</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_DATA_BLOCKS_PER_KEY</td>
<td>N/A</td>
</tr>
<tr>
<td>CLUSTERING_FACTOR</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the indexes: VALID</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>N/A</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>N/A</td>
</tr>
<tr>
<td>DEGREE</td>
<td>N/A</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>N/A</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>N/A</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>N/A</td>
</tr>
<tr>
<td>GENERATED</td>
<td>N/A</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>N/A</td>
</tr>
<tr>
<td>OBJECT_NAME</td>
<td>Name of object</td>
</tr>
<tr>
<td>SUBOBJECT_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>OBJECT_ID</td>
<td>Object number of the object</td>
</tr>
<tr>
<td>DATA_OBJECT_ID</td>
<td>N/A</td>
</tr>
<tr>
<td>OBJECT_TYPE</td>
<td>Type of object</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_DDL_TIME</td>
<td>N/A</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>N/A</td>
</tr>
<tr>
<td>STATUS</td>
<td>State of the object: VALID</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>N/A</td>
</tr>
<tr>
<td>GENERATED</td>
<td>N/A</td>
</tr>
</tbody>
</table>
The user’s private synonyms.

<table>
<thead>
<tr>
<th>SYNONYM_NAME</th>
<th>Name of the synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_OWNER</td>
<td>Owner of the object referenced by the synonym</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Name of the object referenced by the synonym</td>
</tr>
<tr>
<td>DB_LINK</td>
<td>N/A</td>
</tr>
</tbody>
</table>
**USER_TAB_COLUMNS**

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

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>Table, view, or cluster name</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>Column name</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>Datatype of the column</td>
</tr>
<tr>
<td>DATA_TYPE_MOD</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_TYPE_OWNER</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_LENGTH</td>
<td>Maximum length of the column in bytes</td>
</tr>
<tr>
<td>DATA_PRECISION</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_SCALE</td>
<td>Digits to the right of decimal point in a number</td>
</tr>
<tr>
<td>NULLABLE</td>
<td>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.</td>
</tr>
<tr>
<td>COLUMN_ID</td>
<td>Sequence number of the column as created</td>
</tr>
<tr>
<td>DEFAULT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>DATA_DEFAULT</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_DISTINCT</td>
<td>Number of distinct values in each column of the table</td>
</tr>
<tr>
<td>LOW_VALUE</td>
<td>Second lowest and the 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.</td>
</tr>
<tr>
<td>HIGH_VALUE</td>
<td></td>
</tr>
<tr>
<td>DENSITY</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_NULLS</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_BUCKETS</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>N/A</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>N/A</td>
</tr>
<tr>
<td>CHARACTER_SET_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>CHAR_COL_DECL_LENGTH</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### USER_TAB_COMMENTS

Comments on the tables and views owned by the user.

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

### USER_TABLES

Description of the user’s own tables.

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>Name of the table</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLESPACE_NAME</td>
<td>Name of the tablespace containing the table</td>
</tr>
<tr>
<td>CLUSTER_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>IOT_NAME</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_FREE</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_USED</td>
<td>N/A</td>
</tr>
<tr>
<td>INI_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_TRANS</td>
<td>N/A</td>
</tr>
<tr>
<td>INITIAL_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>NEXT_EXTENT</td>
<td>N/A</td>
</tr>
<tr>
<td>MIN_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_EXTENTS</td>
<td>N/A</td>
</tr>
<tr>
<td>PCT_INCREASE</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELISTS</td>
<td>N/A</td>
</tr>
<tr>
<td>FREELIST_GROUPS</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING</td>
<td>N/A</td>
</tr>
<tr>
<td>BACKED_UP</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_ROWS</td>
<td>Number of rows in the table</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### USER_TABLES

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPTY_BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_SPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>CHAIN_CNT</td>
<td>N/A</td>
</tr>
<tr>
<td>AVG_ROW_LEN</td>
<td>Average length of a row in the table in bytes</td>
</tr>
<tr>
<td>AVG_SPACE_FREELIST_BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>NUM_FREELIST_BLOCKS</td>
<td>N/A</td>
</tr>
<tr>
<td>DEGREE</td>
<td>N/A</td>
</tr>
<tr>
<td>INSTANCES</td>
<td>N/A</td>
</tr>
<tr>
<td>CACHE</td>
<td>N/A</td>
</tr>
<tr>
<td>TABLE_LOCK</td>
<td>N/A</td>
</tr>
<tr>
<td>SAMPLE_SIZE</td>
<td>N/A</td>
</tr>
<tr>
<td>LAST_ANALYZED</td>
<td>N/A</td>
</tr>
<tr>
<td>PARTITIONED</td>
<td>N/A</td>
</tr>
<tr>
<td>IOT_TYPE</td>
<td>N/A</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>N/A</td>
</tr>
<tr>
<td>NESTED</td>
<td>N/A</td>
</tr>
<tr>
<td>BUFFER_POOL</td>
<td>N/A</td>
</tr>
</tbody>
</table>
**USER USERS**

Information about the current user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>Name of the user</td>
</tr>
<tr>
<td>USER_ID</td>
<td>N/A</td>
</tr>
<tr>
<td>ACCOUNT_STATUS</td>
<td>N/A</td>
</tr>
<tr>
<td>LOCK_DATE</td>
<td>N/A</td>
</tr>
<tr>
<td>EXPIRY_DATE</td>
<td>N/A</td>
</tr>
<tr>
<td>DEFAULT_TABLESPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>TEMPORARY_TABLESPACE</td>
<td>N/A</td>
</tr>
<tr>
<td>CREATED</td>
<td>N/A</td>
</tr>
<tr>
<td>EXTERNAL_NAME</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**USER_VIEWS**

Text of views owned by the user.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIEW_NAME</td>
<td>Name of the view</td>
</tr>
<tr>
<td>TEXT_LENGTH</td>
<td>Length of the view text</td>
</tr>
<tr>
<td>TEXT</td>
<td>First line of view text</td>
</tr>
<tr>
<td>TYPE_TEXT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>TYPE_TEXT</td>
<td>N/A</td>
</tr>
<tr>
<td>OID_TEXT_LENGTH</td>
<td>N/A</td>
</tr>
<tr>
<td>OID_TEXT</td>
<td>N/A</td>
</tr>
<tr>
<td>VIEW_TYPE_OWNER</td>
<td>N/A</td>
</tr>
<tr>
<td>VIEW_TYPE</td>
<td>N/A</td>
</tr>
</tbody>
</table>
accessing
   AS/400 file members, 7-7
Advanced Security, 1-6
agent trace files, 1-11, A-3
ALL_CATALOG view, C-3
ALL_COL_COMMENTS view, C-3
ALL_CONSTRAINTS view, C-4
ALL_DB_LINKS data dictionary view, 7-4
ALL_IND_COLUMNS view, C-5
ALL_INDEXES view, C-5
ALL_OBJECTS view, C-7
ALL_SYNONYMS view, C-7
ALL_TAB_COLUMNS view, C-8
ALL_TAB_COMMENTS view, C-9
ALL_TABLES view, C-9
ALL_USERS view, C-10
ALL_VIEWS view, C-11
ALTER SESSION CLOSE DATABASE LINK
   statement, 7-3
ANO, 1-5
API, 5-2
APPC/LU6.2
   protocol, 1-4
application development
   database link, 9-2
application server support, 1-5
applications
   SQL*Plus, 1-8
applications program interface See API
architecture
   gateway, 1-9
array
   definition, 9-3
   processing, 6-18, 9-3
ARRAY BLOCK SIZE parameter, 6-18, 9-5
AS/400 database
   accessing file members, 7-7
   AUTOCREATE CONTROLLER parameter, 5-8
B
   binary literal notation, restriction, 2-7
   bind variables, 9-27
      restriction, 2-7, 2-8
   BLOCKSIZE data area, 6-18
C
   CCSID, 6-19, B-3
   and NLS, B-3
      supported character sets, B-3
      supported languages, B-5
      supported territories, B-5
   CFGTCP command, 5-17
   changes to the system upon installation, 1-11
   CHAR format, 9-21
   CHARACTER SET parameter, B-6
   character strings
      converting datatypes, 9-20
      processing, 9-19
   CHECKSUM command, 1-6
   CHGGTWDBG command
      command description, 6-14
      purpose, 6-2
   CHGORANET command
      assigning a port number, 5-10
changing parameters, A-2
purpose, 6-2
trace files, 5-5
CHGORAPJE command
command description, 6-10
prestarted jobs, 6-7
purpose, 6-2
CHGORATUN command, B-2
changing default BLOCKSIZE, 6-18
command description, 6-10
data conversion, 6-19
gateway configuration, B-2
language specification, B-2
number of DB2/400 cursors, 7-5
purpose, 6-2
CHGRECOPRF command, 6-7
cautions to synchronize recovery
password, 6-17, 8-7
command description, 6-16
CHGUSRPRF command, 6-7
clauses
CONNECT BY, 9-25
CONNECT TO, 8-6
FOR UPDATE, 9-25, 9-26
USING, 7-2
WHERE, 9-25
coded character set identifier, See CCSID
coeexistence between gateways, 4-2
COLUMN_PRIVILEGES view, C-11
columns
DECIMAL, 9-22
INTEGER, 9-22
NUMERIC, 9-22
PACKED, 9-22
ROUTINE_NAME, 9-9
ROUTINE_SCHEMA, 9-9
SMALLINT, 9-22
command summary, gateway commands, 6-2
commands
CFGTCP, 5-17
CHGGTWDBG
command description, 6-14
purpose, 6-2
CHGORANET
assigning port number, 5-10
command description, 6-8
purpose, 6-2
trace files, 5-5
CHGORAPJE
command description, 6-10
purpose, 6-2
CHGORATUN
changing BLOCKSIZE, 6-18
command description, 6-10
data conversion, 6-19
gateway configuration, B-2
language specification, B-2
purpose, 6-2
CHGRECOPRF
cautions to synchronize recovery
password, 6-17, 8-7
CREATE COMMAND, 4-6
CREATE SNAPSHOT, 7-14
CREATE TABLE, 7-14
CRTLFE, 7-7
CRTLINET, 5-8
CRTLINSIDL, 5-8
CRTLINTRN, 5-8
CRTLNX25, 5-8
CRTORADDB
command description, 6-15
purpose, 6-2
CRTORAGTWI
purpose, 6-2
CRTUSRPFR, 1-12
DSPLIND, 5-8
DSPPGM, 4-12
ENDSBS, 8-5
EXECUTE, 1-7
INSERT, 7-14
JOIN, 9-3
ORAMON, 1-10, 8-2
PING, 5-12
SBMJOB, 4-14
SQL, 7-3
SQL*Plus
COPY, 7-12, 7-13, 7-15
DELETE, 9-27
DESCRIBE, restrictions, 2-2
INSERT, 7-13
STORALS N, 5-12
STRSSB, 1-10
commit confirm protocol, 1-9
COMMIT_POINT_STRENGTH setting, 7-10
commitment control, 4-11, 9-10
concurrent connections
TCP/IP, memory, 3-2
CONNECT BY clause, 9-25
CONNECT BY clause, not supported in SQL, 2-8
CONNECT TO clause, 7-2, 7-4, 7-6, 8-6
statement, 7-4
user ID, 7-4
convert
SQL to make compatible with DB2/400 server, 1-13
COPY command, 7-12, 7-13, 7-15
copying data
from the DB2/400 server, 7-14
to DB2/400 server, 7-12
copying the gateway, CRTORAGTWI command, 6-4
COUNT function, 9-27
CREATE COLLECTION command, 4-6
CREATE DATABASE LINK command, 7-2
CREATE SNAPSHOT command, 7-14
CREATE TABLE command, 1-7, 7-14
CRTLF command, 7-7
CRTLINETH command, 5-8
CRTLINSDL DLC command, 5-8
CRTLINTRN command, 5-8
CRTLINX25 command, 5-8
CRTORADDB command
command description, 6-15
purpose, 6-2
CRTORAGTWI command, 6-4
command description, 6-4
purpose, 6-2
CRTUSRPRF command, 1-12
cursor stability, 4-11
cursors for DB2/400, 7-5

D

data control language, See DCL
data conversion, 6-19
data definition language, See DDL
data dictionary, 4-8, 4-11, 6-7
creating, 6-15
emulation in DB2/400 server, 9-28
tables
restriction with v4 gateways, 2-5
using, 9-28
views, 7-5, 9-28
ALL_CATALOG, C-3
ALL_COL_COMMENTS, C-3
ALL_CONSTRAINTS, C-4
ALL_DB_LINKS, 7-4
ALL_INDEXES, C-5
ALL_OBJECTS, C-7
ALL_SYNONYMS, C-7
ALL_TAB_COLUMNS, C-8
ALL_TAB_COMMENTS, C-9
ALL_TABLES, C-9
ALL_USERS, C-10
ALL_VIEWS, C-11
COLUMN_PRIVILEGES, C-11
DBA_DB_LINKS, 7-5
OTGREGISTER, 9-28, C-12
TABLE_PRIVILEGES, C-13
USER_CATALOG, C-13
USER_COL_COMMENTS, C-13
USER_CONS_COLUMNS, C-14
USER_CONSTRAINTS, C-14
USER_DB_LINKS, 7-4
USER_INDEXES, C-15
USER_OBJECTS, C-16
USER_SYNONYMS, C-17
USER_TAB_COLUMNS, C-18
USER_TAB_COMMENTS, C-19
USER_TABLES, C-19
USER_USERS, C-21
USER VIEWS, C-21
DATABASE DOMAIN parameter, 6-11
database link, 7-2
accessing data, 7-4
behavior, 7-2
closing, 7-3
connecting to DB2/400 databases, 9-2
creating, 7-2
data dictionary views
  ALL_DB_LINKS, 7-4
  DBA_DB_LINKS, 7-5
  USER_DB_LINKS, 7-4
dropping links, 7-4
examining, 7-4
function in the gateway, 1-11
limiting, 7-5
using Oracle Net, 7-3
database triggers, 1-5
datatypes
conversion
  for Oracle tools, 9-15
  from DB2/400 to Oracle, B-6
  Oracle to DB2/400 datatypes, 9-15
  converting character strings, 9-20
  DATE, 9-20
  DB2/400 GRAPHIC, 9-17
  IBM DATE, 9-21
  LONG, 9-20
    known restriction, 2-9
  LONG VARCHAR, 9-20
  Oracle DATE, 9-21
  restrictions, 9-15
  TIME, 9-20, 9-21
  CHAR format, 9-21
  TIMESTAMP, 9-20
    CHAR format, 9-21
  VARCHAR, 9-20
  zoned decimal, 9-28
date and time operations
  Oracle and DB2/400, 9-20
date arithmetic
  restriction, 2-5
  DATE datatype, 9-20
dates
  21st century, 9-22
  TO_DATE function, 9-22
  DB_DOMAIN parameter, Oracle9 Database Server, 2-7
  DB2/400
  Application Server, 1-10
cursors, 7-5
database
  read, 9-2
  write, 9-2
differences from Oracle server, 9-27
  executing stored procedures, 9-8
  GRAPHIC datatype support and NCHAR, 9-17
  native SQL, 1-7
  native stored procedures, 1-8, 9-7

DB2/400 server
compatibility
  with SQL form, 1-13
  triggers, 7-12
  DB2/400 VALUE function, 1-13
  DBA_DB_LINKS data dictionary view, 7-5
  DBMS_HS_PASSTHROUGH.EXECUTEImmediate function, 9-11

DCL
  issue native DCL from Oracle environment, 1-7
DD basic tables
  restriction, 2-7
DDL
  Oracle applications use native DDL, 1-7
debugging parameters
  changing, 6-14
  DECIMAL column, 9-22
  DECnet protocol, 1-4
definitions, 5-3
DELETE command, 9-27
DESCRIBE SQL*Plus command, 2-2
developing applications, 9-2
diagnosing errors, 10-4
dictionary mapping, 1-6
distributed database
  Oracle Net, 5-2
distributed processing
  Oracle Net, 5-2
distributed queries, 7-8
documentation
  errors, 10-5
  requirements, 3-3
domain name, 5-11
double-byte character support, 9-17
DROP DATABASE LINK statement, 7-4
DSPLIND command, 5-8
DSPPGM command, 4-12

E
EMPDB2 synonym, 7-5
ENDSBS command, 8-5
error
1017, invalid user ID or password, 6-17, 8-7
ORA-12154 connection error, 5-15
ORA-2068 connection error, 5-15
ORA-2070, 2-9
ORA-28500
  file not journalled, 4-13
ORA-28500 connection problems, 5-15
ORA-28509 connection error, 5-15
ORA-28511 connection error, 5-15
ORA-3114 connection error, 5-15
SQL0105, "Mixed or Graphic String Constant Not Valid.", 9-17

F
fetch reblocking, 9-4
file
  INIT.ORA, 2-7, 6-11
  TNSNAMES.ORA, 5-14, 7-2
file members
  accessing, 7-7
  flat files
    accessing, 7-8
FOR BIT DATA option, 2-7
FOR UPDATE clause, 9-25, 9-26
NOWAIT option, 9-26
FORCE_SBCS, string to force double-byte character support, 9-19
functions
  converting Oracle9 Database Server NVL function to VALUE, 1-13
  COUNT, 9-27
  DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE, 9-11, 9-13
  NVL, 1-13
  SQL
    column, 9-23
    compensated, 9-24
    scalar, 9-23
    translated, 9-24
  TO_DATE, 9-22
  VALUE, 9-24

G
gateway, 9-7
  advantages, 1-3
  access, 7-6
  migration and coexistence, 1-9
  multisite transactions, 1-8
  security, 1-9
  site autonomy, 1-9
  two-phase commit, 1-8
  architecture, 1-9
  array processing, 9-3
  changes made to the OS/400 system, 1-11
  co-existence with previous gateways, 4-2
  commands, 6-2
    CHGGTWDBG, 6-14
    CHGORANET, 6-8, B-2
  changing parameters, A-2
    CHGORAPJE, 6-10
    CHGORATUN, 6-10, 6-18, B-2
    CHGRECOPRF, 6-7
    CHGUSRPRF, 6-7
  command summary, 6-2
  CRTORAGTWI, 6-4
  menu to access, 6-4
  running, 6-3
Index-6

components
  DB2 server, 1-10
  Oracle9 Database Server, 1-10
transparent gateway, 1-10
development applications, 9-2
how it works, 1-10
jobs, 7-7
known restrictions, 2-5
library files, A-2
monitoring, 8-1
multiple copies, 6-4
NLS configuration, B-2
post-installation, 4-11
raising priority level, 4-14
removing observability, 4-14
retrieving data, 6-18
security, 8-6
setting optional parameters, 6-18
starting, 8-5
stopping, 8-5
transparency levels, 1-4
two-phase commit, 7-9
using Oracle stored procedures, 9-5
verifying the version, 4-12
gateway call stack
displaying, 8-6
gateway initialization parameters, 6-11
gateway job locks, 8-6
gateway monitor
  see Oracle gateway monitor
GATEWAY NATIONAL LANGUAGE parameter, 6-11
GENERAL linkage convention, 9-10
GENERAL WITH NULLS linkage convention, 9-10
GLOBAL_NAMES
  known restrictions, 2-7
GO CMDORAGTW, 6-3
GRAPHIC datatype, B-6
  support, B-6

H
hanging condition, connection problems, 5-15
hardware requirements, 3-2
heterogeneous services, 1-3

initialization parameters
  known restrictions for DB2/400, 2-7
host name, 5-11
HS= parameter, 5-13
HS_DB_DOMAIN
  Heterogeneous Services initialization parameter, 2-7
HS_RPC_FETCH SIZE parameter, 9-4
HS_RPC_FETCH_REBLOCKING parameter, 9-4
IBM DATE datatype, 9-21
implicit protocol conversion, 1-5
incorrect output errors, 10-6
initialization parameters
  changing, 6-10
  GLOBAL_NAMES, 2-7
INIT.ORA file, 2-7, 6-11
INSERT
  clause, 9-26
  command, 7-14
  command, restriction, 2-5
installation
  before installing, 4-4
  checking distribution kit, 3-3
  completion, 4-10
  mount product tape, 4-6
  required parameters, 4-6
  set required parameters, 4-6
  start Oracle subsystem, 4-10
  start process, 4-6
  starting Oracle subsystem, 4-10
  verify, 4-9
INTEGER column, 9-22
Integrated File System
  system changes when installing the gateway, 1-11
internal process communication, 5-9
internet support, 1-5
isolation level, 4-11, 4-13, 9-13

J
job locks, 8-6
job logs, 4-12
JOBCTL special authority, 4-5
jobs
  viewing, 7-7
JOIN capability, 1-4
JOIN command, 9-3
journaling
  AS/400 files, 4-12
  definition, 4-11
  error messages, 9-13
  new tables, 9-13
  ORACLE2PC file, 4-13
  requirements, 4-11
Julian dates, restriction, 2-6

K
known problems, 2-4
known restrictions
  date arithmetic, 2-5
DB2/400
  GLOBAL_NAMES initialization parameter, 2-7
DB2/400 components, 2-7
OCI, 2-7
snapshots, 2-5
SQL*Plus DESCRIBE command, 2-2

L
Languages, 1-8
libraries
  new tables, 9-13
  ORASRVLIB, 1-12
  QUSRSYS, 1-12
library files, A-2
listener, 5-3
  client connection, 5-4
  prestarted jobs, 5-4
  TNS, 5-3
literal values, restriction, 2-7
local domain name, 5-11
local host name, 5-11
LOG file members, A-2
logical file members, 7-7
LONG datatype, 9-20
  restriction, 2-9
LONG VARCHAR datatype, 9-20

M
mapping error messages, 10-2
mass delete
  from a segmented tablespace, 9-27
member names
  format, 5-5
members
  accessing AS/400 file members, 7-7
  LOG file, A-2
  ORA file, A-2
  SQLNET, 5-5
  TRC file, A-3
messages
  and error code processing, 10-2
  diagnosing, 10-4
  error, 10-1
  journaling, 9-13
  ORA-00947, 10-4
  ORA-2070, 2-9
Metalink Note 136294.1, connection problems, 5-15
missing functionality, 10-7
Mobile Agents, 1-6
monetizing the gateway, 8-1
  overview, 8-2
  starting Oracle gateway monitor, 8-2
multiple rows
  retrieving, 6-18
multisite transactions, 1-8

N
National Language Support, See NLS
network parameters
  changing, 6-8
NLS
  CCSID, B-3
    supported character sets, B-3
    supported languages, B-5
    supported territories, B-5
gateway commands
Index-8

CHGORANET, B-2
CHGORATUN, B-2
gateway configuration, B-2
CHGORATUN language specification, B-2
message availability, B-6
Oracle server and client configuration, B-6
overview, B-2
notational conventions, xv
NOWAIT option, 9-26
null values
   with stored procedures, 9-10
numeric column, 9-22
numeric datatype, 9-22
NVL function, 1-13

O

observability, 4-14
OCI
   restriction, 2-7
OPEN_LINKS parameter, 7-5
operations
   date and time, 9-20
ORA file members, A-2
ORA(LISTENER) file, 6-9, A-2
ORA(LISTENER_CHG) file, A-2
ORA(LISTENLOCK) file, A-2
ORA(SQLNET) file, 6-9, A-2
ORA-00947 error message, 10-4
ORA-12154 connection error, 5-15
ORA-2068 connection error, 5-15
ORA-2070 message, 2-9
ORA-28500 error, 5-15
   file not journaled, 4-13
ORA-28509 connection error, 5-15
ORA-28511 connection error, 5-15
ORA-3114 connection error, 5-15
Oracle Call Interface See OCI
Oracle DATE datatype, 9-21
Oracle Developer
   Forms
      compatibility, 9-29
Oracle gateway monitor
   display gateway call stack, 8-6
   display open files, 8-6
ORAMON command, 8-2
overview, 8-2
starting, 8-2
starting and stopping gateway, 8-5
starting TCP/IP listener, 8-6
working with gateway job locks, 8-6
working with gateway jobs, 8-5
Oracle integrating server
   and GLOBAL_NAMES initialization
      parameter, 2-7
      function
         regarding SQL, 1-13
Oracle Net, 1-8, 4-14, 5-1
API, 5-2
communications, 5-3
configuring APPC/LU6.2
   define physical line, 5-8
   checklist, 5-10
   define local host name, 5-11
   modify port number, 5-10
   start the listener, 5-12
   verify host name, 5-12
   connecting TCP/IP, 5-12
      add connect descriptor, 5-13
      checklist, 5-7, 5-12
      specify SID names, 5-14
   connections, 5-3
   creating database links, 7-3
   distributed database, 5-2
   distributed processing, 5-2
   file names, 5-5
   listener, 5-3
   LOG file members, A-2
   ORA file members, A-2
   overview, 5-2
   terminology, 5-3
      client, 5-3
      driver, 5-3
      host, 5-3
      network, 5-3
      protocol, 5-3
   tracing job members, 5-5
   TRC file members, A-3
Oracle Precompilers
Index-9

restriction, 2-7
Oracle subsystem
   error reporting, 10-6
Oracle Support Services
   contacting, 10-4
ORACLE2PC file, 4-13, 7-10
ORACLE2PC table, 1-8
Oracle9 Database Server, 1-4, 6-11
   and array size, 9-4, 9-5
   and database links, 1-11
   and error detection, 10-4
copying data from the DB2/400 server, 7-14
copying data to DB2/400 server, 7-12
database triggers, 1-5
date and time operations, 9-20
differences from DB2/400, 9-27
distributed capabilities, 1-4
distributed query optimization, 1-5
mapping DB2/400 error messages, 10-2
performing character string operations, 9-19
post-processing of SUBSTR SQL function, 2-8
stored procedures, 1-5
triggers, 7-11, 7-12
two-phase commit protection, 1-5
ORADBMBOPT Data Area (or Environment
   variable), 9-19
ORAMON command, 1-10, 8-2
ORARAW
   data area, 6-18
ORASRVLIB library
   creation, 1-12
OTGREGISTER view, 9-28, C-12
outer joins, SQL functions, 1-4
overview of Oracle Net, 5-2

P
PACKED column, 9-22
panel
   Change Oracle Gateway Debugging
      Option, 6-14
   Change Oracle Gateway Initialization Parameters
      (page 1), 6-12
   Change Oracle Gateway Initialization Parameters
      (page 2), 6-13
   Change Oracle Network Parameters, 6-8
   Change Recovery Profile Parameters, 6-16
   CMDORAGTW Oracle Commands, 6-3
   Create Oracle Prestart Parameters, 6-10
   Create Oracle Transparent Gateway, 6-5, 6-6
parameters
   ARRAY BLOCK SIZE, 6-18, 9-5
   AUTOCREATE CONTROLLER, 5-8
data dictionary, 6-15
DATABASE DOMAIN, 6-11
DB_DOMAIN
   Oracle9 server, 2-7
displaying, 6-14
HS_DB_DOMAIN, Heterogeneous Services
   initialization parameter, 2-7
HS_PRC_FETCH_REBLOCKING, 9-4
HS_RPC_FETCH SIZE, 9-4
initialization, 6-10
network, 6-8
OPEN_LINKS, 7-5
   optional, 6-18
   READONLY, 6-19
   required, 4-6
RPC FETCH REBLOCKING, 6-11, 9-4
RPC FETCH SIZE, 6-11, 9-4
setting, 4-6, 6-18
SQLNET.ORA, 5-5
pass-through, 1-7, 9-11
   examples, 9-12
   function, 9-11
   result sets, 9-11
   example, 9-14
   retrieving result sets, 9-14
performance issues, 10-7
PING command, 5-12
PL/SQL
   routine, 1-7
   standard Oracle, 1-8
port numbers
   changing them, 5-10
post-installation
   configure Oracle Net, 4-14
   journal ORACLE2PC file, 4-13
   raise priority level, 4-14
remove observability, 4-14
steps, 4-11
verify journaling, 4-12
post-processing, 1-13
precompiler limitations, 2-7
prestarted jobs
  changing parameters, 6-10
priority level, 4-14
processing SQL, 9-25
processor requirement, 3-2
product set, 2-2
protocols
  APPC/LU6.2, 1-4
  commit confirm, 1-9
  DECnet, 1-4
  definition, 5-3
  SPX/IPX, 1-4, 1-5
  TCP/IP, 1-4, 1-5
remote transaction program, 3-2
replication
  heterogeneous environment, 7-11
requirements
  documentation, 3-3
  hardware, 3-2
  software, 3-3
  journaling, 4-11
result sets, 9-11
retrieving
  multiple rows, 6-18, 9-3
ROUTINE_NAME column, 9-9
ROUTINE_SCHEMA column, 9-9
routines
  PL/SQL, 1-7
ROWID
  not supported, 2-8
RPC FETCH REBLOCKING parameter, 6-11, 9-4
RPC FETCH SIZE parameter, 6-11, 9-4
running gateway commands, 6-3
RUNORALSN program runs the TCP/IP
  listener, A-2
savepoints, restriction, 2-9
SBMJOB command, 4-14
QSECOFR ID, 4-14
SECADM special authority, 4-4
security
  advanced, 1-6
  ANO, 1-6
  encryption, 1-6
  on the gateway, 8-6
segmented tablespace
  mass delete from, 9-27
SELECT FOR DELETE clause, 9-26
SELECT FOR INSERT clause, 9-26
SELECT FOR UPDATE clause, 9-26
SELECT statement, 1-13, 9-3, 9-4, 9-25
  array size in fetch reblocking, 9-4
session connection, 7-2
SID names, 5-14
SMALLINT column, 9-22
snapshots
complete refresh, 7-14
restriction, 2-5
CREATE SNAPSHOT command, 7-14
fast refresh, 2-5
restriction, 2-9
feature capability, 7-11
software requirements, 3-3
special authority
*JOBCTL, 4-5
*SECADM, 4-4
special registers in DB2/400, 9-28
SPX/IPX protocol, 1-4, 1-5
SQL, 1-7
ANSI standard, 1-7
commands, 7-3
compatibility with gateway through Oracle
integrating server, 1-13
constructs, Oracle-specific, 9-25
conversion for compatibility with DB2/400
server, 1-13
differences, 1-13
functions
column, 9-23
compatibility, 9-23
compensated, 9-24
scalar, 9-23
translated, 9-24
ISO standard, 1-7
known restrictions, 2-8
limitations, 2-8
passthrough, 9-11
statements, 9-2
passing through gateway, 9-11
SQL construct processing, 9-25
SQL*Plus, 1-8
commands
COPY, 7-12, 7-13
DELETE, 9-27
DESCRIBE, 2-2
INSERT, 7-13
SQL0105 error, "Mixed or Graphic String Constant Not Valid.", 9-17
SQLNET member, 5-5
SQLNET(ORA) file, 5-5
SQLNET.ORA parameter, 5-5
starting the gateway, 8-5
statements
ALTER SESSION CLOSE DATABASE
LINK, 7-3
CONNECT TO, 7-4
CREATE DATABASE LINK, 7-2
DB2
CREATE TABLE, 1-7
DROP DATABASE LINK, 7-4
SELECT, 9-3, 9-4, 9-25
SQL, 9-2
passing through gateway, 9-11
stopping the gateway, 8-5
storage measurements, xv
stored procedures, 1-5, 1-7
columns
ROUTINE_NAME, 9-9
ROUTINE_SCHEMA, 9-9
considerations, 9-10
DB2/400 with the gateway, 9-7
execution of DB2/400 procedures from
applications, 9-8
linkage conventions, 9-10
native DB2, 1-8
Oracle, 1-7, 9-5
PL/SQL, 9-7
synonyms for, 9-6
two-phase commit, 9-7
using, 9-5
STORALSN command, 5-12
STRSBS command, 1-10
Structured Query Language See SQL
support
contacting, 10-4
synonym feature, 7-5
syntax
conventions used in this book, xv

T

TABLE_PRIVILEGES view, C-13
TCP/IP
AS/400 database, 5-3
bequeath mechanism, 5-9
concurrent connections, memory, 3-2
configuring Oracle Net, 5-9
connecting to the AS/400, 5-12
connection flow, 5-9
connection problems
  troubleshooting, 5-15
  inherit mechanism, 5-9
  local domain name, 5-11
  local host name, 5-11
  port numbers, 4-8, 5-10, 6-7
  protocol, 1-4, 1-5
  socket interface, 4-5
  starting the listener, 8-6
TCP/IP listener, 1-10
  starting, 8-6
terminology, Oracle Net
  client, 5-3
  driver, 5-3
  host, 5-3
  network, 5-3
  protocol, 5-3
TIME datatype, 9-20
  CHAR format, 9-21
TIMESTAMP datatype, 9-20
  CHAR format, 9-21
TNS
  connect descriptor, 7-3
TNSNAMES.ORA file, 5-13, 5-14, 7-2
  connect descriptor, 7-2
TNSPING command, 5-17
TO_CHAR function, 2-6
TO_DATE function, 2-6, 9-21, 9-22
trace files, 5-5
  creating, 1-11
  tracing, 5-6
transparency
  access method, 1-4
  data storage, 1-4
  location, 1-4
  network, 1-4
  operating system, 1-4
TRC file members, 5-6, A-3
triggers, 7-11
  DB2 server, 7-12
  Oracle9 Database Server, 7-12
twenty-first century dates, 9-22
two-phase commit, 1-5, 1-8, 9-7
  description, 7-9
  recorded in, 7-10
  transaction recovery, 7-11
  unsupported statement, 7-10

U

UNION capability, 1-4
user class
  QSECOFR, 4-4
  USER, 4-5
user IDs
  CONNECT TO, 7-4
USER_CATALOG view, C-13
USER_COL_COMMENTS view, C-13
USER_CONS_COLUMNS view, C-14
USER_CONSTRAINTS view, C-14
USER_DB_LINKS view, 7-4
USER_INDEXES view, C-15
USER_OBJECTS view, C-16
USER_SYNONYMS view, C-17
USER_TAB_COLUMNS view, C-18
USER_TAB_COMMENTS view, C-19
USER_TABLES view, C-19
USER_USERS view, C-21
USER_VIEWS view, C-21
USING clause, 7-2

V

VALUE function, 9-24
VARCHAR datatype, 9-20
views
data dictionary
  ALL_CATALOG, C-3
  ALL_COL_COMMENTS, C-3
  ALL_CONSTRAINTS, C-4
  ALL_DB_LINKS, 7-4
  ALL_IND_COLUMNS, C-5
  ALL_INDEXES, C-5
  ALL_OBJECTS, C-7
  ALL_SYNONYMS, C-7
  ALL_TAB_COLUMNS, C-8
  ALL_TAB_COMMENTS, C-9
ALL_TABLES, C-9
ALL_USERS, C-10
ALL_VIEWS, C-11
COLUMN_PRIVILEGES, C-11
DBA_DB_LINKS, 7-5
OTGREGISTER, 9-28, C-12
TABLE_PRIVILEGES, C-13
USER_CATALOG, C-13
USER_COL_COMMENTS, C-13
USER_CONS_COLUMNS, C-14
USER_CONSTRAINTS, C-14
USER_DB_LINKS, 7-4
USER_INDEXES, C-15
USER_OBJECTS, C-16
USER_SYNONYMS, C-17
USER_TAB_COLUMNS, C-18
USER_TAB_COMMENTS, C-19
USER_TABLES, C-19
USER_USERS, C-21
USER_VIEWS, C-21

W
WHERE clause, 7-15, 9-25
wireless communication, 1-6
writing to DB2/400 database, 9-3
WRKTRCLOG command, A-3

Z
zoned decimal operations, 9-28