

Oracle Transparent Gateway®

Sybase on Sun SPARC Solaris Administrator's Guide

Release 8.1.6

April 2000

Part No. A82869-01

Copyright © 1996, 2000, Oracle Corporation. All rights reserved.

Primary Authors: Stuart Cornfield, Orit Curiel, Julian Weiss

Contributors: Esti Adan, Jacco Draaijer, Pavana Jain, Cynthia Kibbe, Kishan Peyetti

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

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

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

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

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

Oracle, Oracle Open Gateways, Oracle Procedural Gateway, Oracle Transparent Gateway, SQL*Net, and SQL*Plus are registered trademarks, and Oracle8i, Oracle8, Oracle7, Oracle Discoverer, PL/SQL, and Net8 are trademarks of Oracle Corporation. All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Contents

Send Us Your Comments	xiii
Preface.....	xv
Audience.....	xvi
Related Publications	xvi
Notational Conventions	xvii
Your Comments are Welcome.....	xviii
 1 Release Information	
Product Set	1-2
System Requirements	1-2
Hardware Requirements	1-3
Software Requirements.....	1-3
Tested Configurations	1-4
Database Compatibility Issues for Sybase	1-5
Chained Mode.....	1-5
Column Definitions.....	1-5
Naming Rules.....	1-6
Rules for Naming Objects	1-6
Case Sensitivity	1-6
Data Types	1-7
Binary Literal Notation.....	1-7
Data Type Conversion.....	1-8

Queries	1-8
Row Selection	1-8
Empty Strings.....	1-8
Empty Bind Variables	1-9
Locking.....	1-9
Known Restrictions	1-9
Transactional Integrity	1-10
Transaction Capability	1-11
COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors	1-11
Stored Procedures.....	1-11
Pass-Through Feature	1-11
Sybase NCHAR and NVARCHAR Data Types	1-12
SQL Syntax.....	1-12
SELECT FOR UPDATE Statement	1-12
WHERE CURRENT OF Clause	1-12
CONNECT BY Clause.....	1-12
Subqueries in UPDATE Statement.....	1-12
ROWID.....	1-13
Subqueries in INSERT Statement.....	1-13
SQL*Plus COPY Command with Lowercase Table Names	1-13
Database Links	1-13
National Language Support.....	1-14
Known Problems.....	1-14
FLOAT Data Type.....	1-15
VARBINARY Data Type.....	1-15
Encrypted Format Login.....	1-15
Aggregate Function with CREATE TABLE or CREATE VIEW Statement	1-15
Date Arithmetic.....	1-16
Sybase IMAGE and TEXT Data Types.....	1-16
String Functions	1-17
Schema Names and PL/SQL	1-17
Data Dictionary Views and PL/SQL	1-17
Stored Procedures.....	1-17

2 Introduction

Oracle Heterogeneous Services	2-2
Oracle Transparent Gateway for Sybase	2-2
Gateway Architecture	2-3
Gateway Process Flow	2-5
Additional Features	2-6
Remote Data Access	2-7
Elimination of Unnecessary Data Duplication.....	2-7
Heterogeneous Database Integration	2-7
Application Development and End User Tools	2-7
Two-Phase Commit and Multi-Site Transactions.....	2-8
Query Optimization	2-8
Error Mapping and Logging.....	2-8
Stored Procedure Execution.....	2-9
Pass-Through Feature	2-9

3 Installing the Gateway

Before You Begin	3-2
Determine the Configuration of the Gateway Components	3-2
Installation Worksheet	3-4
Installing the Gateway from the CD-ROM	3-4
De-installing the Gateway	3-8

4 Configuring the Gateway

Configuring the Gateway	4-2
Task 1: Choose a System ID for the Gateway.....	4-2
Task 2: Customize the Initialization Parameter File.....	4-2
Configuring Net8 Listener for the Gateway	4-3
Task 1: Configure Net8 TNS Listener for the Gateway.....	4-3
Example of Address to Listen On in listener.ora File	4-4
Example of Gateway to Start in listener.ora File	4-4
Task 2: Stop and Start the TNS Listener for the Gateway	4-5
Configuring the Oracle Database Server for Gateway Access	4-7
Configuring Oracle Net8 for the Oracle Database Server	4-7

Configuring tnsnames.ora.....	4-7
Performing Optional Configuration Tasks	4-9
Configuring for Two-Phase Commit	4-9
Task 1: Create a Recovery Account and Password.....	4-9
Task 2: Create the Transaction Log Table	4-10
Specifying an Owner	4-11
Creating Database Links	4-11
Configuring the Gateway for Multiple Sybase Databases	4-12
Multiple Databases Example: Configuring the Gateway	4-13
Multiple Databases Example: Configuring the Sybase Environment.....	4-14
Multiple Databases Example: Configuring Oracle Net8 Listener	4-14
Multiple Databases Example: Stopping and Starting the TNS Listener	4-15
Multiple Databases Example: Configuring the Oracle Database Server for Gateway Access	4-15
Configuring Net8 on the Oracle Database Server for Multiple Gateway Instances..	4-15
Multiple Databases Example: Accessing Sybase Data	4-16

5 Using the Gateway

Accessing Sybase Data Through the Gateway.....	5-2
Oracle Database Server SQL Construct Processing	5-3
Data Type Conversion.....	5-4
Using Synonyms	5-5
Using the Pass-Through Feature	5-5
Executing Stored Procedures and Functions	5-6
Performing Distributed Queries.....	5-7
Example of a Distributed Query.....	5-7
Two-Phase Commit	5-8
Distributed Sybase Transactions	5-8
Copying Data from the Oracle Database Server to Sybase	5-9
Copying Data from Sybase to the Oracle Database Server	5-10
Monitoring Gateway Sessions	5-10

6 Case Studies

Case Descriptions.....	6-2
CD-ROM Contents	6-2

Demonstration Files	6-2
Demonstration Requirements	6-3
Creating Demonstration Tables	6-3
Demonstration Table Definitions	6-4
Demonstration Table Contents.....	6-5
Case 1: Simple Queries	6-6
Case 2: A More Complex Query	6-7
Case 3: Joining Sybase Tables	6-8
Case 4: Write Capabilities.....	6-9
DELETE Statement	6-9
UPDATE Statement.....	6-10
INSERT Statement	6-11
Case 5: Data Dictionary Query	6-12
Case 6: The Pass-Through Feature.....	6-13
UPDATE Statement.....	6-13
SELECT Statement.....	6-14
Case 7: Executing Stored Procedures	6-16

7 Diagnosing Problems

Problem Identification	7-2
Installation and Configuration Issues	7-3
How do I verify that the installation succeeded?	7-3
Does the Oracle Installer configure everything?	7-3
Net8 Configuration Problems.....	7-3
Common Gateway Problems	7-5
Analyzing Error Messages	7-6
Message and Error Code Processing	7-6
Oracle Database Server Error Codes.....	7-7
Gateway Error Codes and Message Formats	7-8
Tracing.....	7-9
Trace Log Files	7-9
Reviewing the Trace Output.....	7-9
Oracle Support Services	7-10

A Supported SQL Syntax and Functions

Supported SQL Statements	A-2
DELETE.....	A-2
INSERT.....	A-2
SELECT.....	A-2
UPDATE.....	A-3
Oracle Functions	A-3
Functions Not Supported by Sybase.....	A-3
Functions Supported by Sybase.....	A-3
Arithmetic Operators.....	A-4
Comparison Operators.....	A-4
String Functions.....	A-4
Group Functions.....	A-5
Other Functions.....	A-5

B Data Dictionary

Data Dictionary Support	B-2
Sybase System Tables.....	B-2
Accessing the Gateway Data Dictionary.....	B-2
Direct Queries to Sybase Tables.....	B-3
Supported Views and Tables.....	B-3
Data Dictionary Mapping	B-4
Default Column Values.....	B-5
Gateway Data Dictionary Descriptions	B-5

C Heterogeneous Services Initialization Parameters

Initialization Parameter File Syntax	C-2
HS_FDS_CONNECT_INFO	C-3
HS_FDS_DEFAULT_OWNER	C-4
HS_FDS_PROC_IS_FUNC	C-4
HS_FDS_RECOVERY_ACCOUNT	C-4
HS_FDS_RECOVERY_PWD	C-5
HS_FDS_TRACE_LEVEL	C-5
HS_FDS_TRANSACTION_MODEL	C-6

List of Figures

2-1	Gateway Processing	2-4
2-2	Gateway Process Flow	2-5
3-1	Gateway Component Configuration	3-3
5-1	Distributed Access Manager View of a Session	5-11

List of Tables

3-1	Installation Worksheet.....	3-4
3-2	Installation Prompts and Responses.....	3-6
5-1	Data Type Conversions	5-4
7-1	Oracle Support Services Worksheet	7-10
B-1	Gateway Supported Views and Tables	B-3
B-2	Oracle Data Dictionary View Names and Sybase Equivalents	B-4
B-3	ALL_CATALOG.....	B-6
B-4	ALL_COMMENTS.....	B-6
B-5	ALL_CONS_COLUMNS.....	B-6
B-6	ALL_CONSTRAINTS.....	B-6
B-7	ALL_INDEXES	B-7
B-8	ALL_OBJECTS.....	B-9
B-9	ALL_TAB_COLUMNS.....	B-10
B-10	ALL_TAB_COMMENTS.....	B-11
B-11	ALL_TABLES.....	B-11
B-12	ALL_USERS	B-13
B-13	ALL_VIEWS.....	B-13
B-14	DBA_CATALOG.....	B-13
B-15	DBA_COL_COMMENTS.....	B-14
B-16	DBA_OBJECTS	B-14
B-17	DBA_TAB_COLUMNS	B-15
B-18	DBA_TAB_COMMENTS	B-16
B-19	DBA_TABLES	B-16
B-20	DICT_COLUMNS	B-18
B-21	DICTIONARY.....	B-18
B-22	DUAL.....	B-18
B-23	TABLE_PRIVILEGES.....	B-18
B-24	USER_CATALOG	B-19
B-25	USER_COL_COMMENTS	B-19
B-26	USER_CONS_COLUMNS	B-19
B-27	USER_CONSTRAINTS.....	B-19
B-28	USER_INDEXES	B-20
B-29	USER_OBJECTS.....	B-22
B-30	USER_TAB_COLUMNS.....	B-23
B-31	USER_TAB_COMMENTS.....	B-24
B-32	USER_TABLES.....	B-24
B-33	USER_USERS	B-25
B-34	USER_VIEWS.....	B-26

Send Us Your Comments

Oracle Transparent Gateway for Sybase on Sun SPARC Solaris Administrator's Guide, Release 8.1.6

Part No. A82869-01

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

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

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- E-mail: infodev@us.oracle.com
- FAX - (650) 506-7228. Attn: Information Development
- Postal service:
Information Development Documentation Manager
Oracle Corporation
500 Oracle Parkway, Mailstop 4op7
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number below.

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

Preface

Oracle Transparent Gateway for Sybase allows Oracle client applications to access Sybase data through Structured Query Language (SQL). The gateway, with the Oracle database server, creates the appearance that all data resides on a local Oracle database server, even though the data can be widely distributed.

This Preface contains the following sections:

- [Audience](#)
- [Related Publications](#)
- [Notational Conventions](#)
- [Your Comments are Welcome](#)

Audience

Read this guide if you are responsible for the following tasks:

- Installing and configuring Oracle Transparent Gateway for Sybase
- Diagnosing gateway errors
- Using the gateway to access Sybase data

Note: You should understand the fundamentals of transparent gateways and the Sun SPARC Solaris operating system before using this guide to install or administer the gateway.

Related Publications

In addition to this guide, the following publications are related to Oracle Transparent Gateway for Sybase:

- *Getting to Know Oracle8i*
- *Oracle Call Interface Programmer's Guide*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle8i Administrator's Guide*
- *Oracle8i Application Developer's Guide*
- *Oracle8i Concepts*
- *Oracle8i Designing and Tuning for Performance*
- *Oracle8i Distributed Database Systems*
- *Oracle8i Error Messages*
- *Oracle8i National Language Support Guide*
- *Oracle8i Reference*
- *Oracle8i SQL Reference*
- *Net8 Administrator's Guide*
- *SQL*Plus User's Guide and Reference*

Note: Collateral gateway information and other related material, such as white papers, are posted on <http://www.oracle.com/gateways>.

Notational Conventions

The following conventions are used in this guide:

<i>Italic Font</i>	Italic characters indicate that the parameter, variable, or expression in the command syntax must be replaced by a value that you provide.
Monospace Font	Monospace font indicates something the computer displays.
Punctuation	Punctuation in syntax examples, other than square brackets and vertical bars, must be entered as shown.
[]	Square brackets enclose optional items. Do not enter the brackets.
()	Parentheses enclose all Net8 Keyword-Value pairs in connect descriptors. They must be entered as part of the connect descriptor, as in (KEYWORD=value).
	A vertical bar represents a choice of two or more options. You must enter one of the options separated by the vertical bar. Do not enter the vertical bar.
UPPERCASE	Uppercase characters within the text represent parameters.

Your Comments are Welcome

We value and appreciate your comment as an Oracle user and reader of our manuals. As we write, revise, and evaluate our documentation, your opinions are the most important feedback we receive.

You can send comments and suggestions about this reference to the Gateway Development department at the following e-mail address:

infodev@us.oracle.com

If you prefer, you can send letters or faxes containing your comments to:

Information Development Documentation Manager
Oracle Corporation
500 Oracle Parkway, 40p7
Redwood Shores, CA 94065
Fax: (650) 506-7228. Attn: Information Development

Release Information

This chapter contains information specific to this release of the Oracle Transparent Gateway for Sybase, and contains the following sections:

- [Product Set](#)
- [System Requirements](#)
- [Tested Configurations](#)
- [Database Compatibility Issues for Sybase](#)
- [Known Restrictions](#)
- [Known Problems](#)

Product Set

This table lists the versions of the components included on the distribution CD-ROM. All components are at production level.

Product	Version Number
Net8 Server	8.1.6.0.0
Net8 Client	8.1.6.0.0
Net8 Assistant	8.1.6.0.0
Net8 Configuration Assistant	8.1.6.0.0
Oracle Universal Installer	1.7.0.18.0A
Oracle Transparent Gateway for Sybase	8.1.6.0.0

System Requirements

This section describes the following:

- [Hardware Requirements](#)
- [Software Requirements](#)

Note: The gateway components can be located on one platform or distributed over several platforms. You should determine the gateway configuration before you determine the hardware and software requirements of the system.

As Oracle continues to support new releases and changes of the Oracle database server and Sybase, the supported configuration will be updated. For current supported configuration information, refer to <http://www.oracle.com/gateways>.

See Also: ["Gateway Process Flow"](#) on page 2-5.

Hardware Requirements

The following table summarizes the hardware requirements for the Oracle Transparent Gateway for Sybase.

Hardware Requirements	
Processor	A Sun SPARC Solaris workstation running the required version of Sun SPARC Solaris
Memory	2.63MB of real memory is recommended for the first user to support the gateway. Each concurrent use of the gateway requires 0.26MB. The total real memory requirement for the concurrent use of the gateway also depends on these factors: <ul style="list-style-type: none">■ The SQL statement issued by the user■ The number of cursors currently opened against Sybase■ The number of columns in the table being accessed
CD-ROM Drive	An internal or external CD-ROM drive
Disk Space	150MB of free disk space

Software Requirements

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

The following table summarizes the software requirements for the Oracle Transparent Gateway for Sybase.

Software Requirements	
Operating System	Sun SPARC Solaris Version 2.6 or 2.7
Oracle database server	Oracle8i Enterprise Edition Server Version 8.1.6 Oracle database server can reside on any supported platform

Software Requirements

Oracle Networking	<p>On gateway machine:</p> <ul style="list-style-type: none">■ Net8 Server Version 8.1.6■ Oracle Adapter for Named Pipes, SPX, or TCP/IP <p>On the Oracle database server machine:</p> <ul style="list-style-type: none">■ Oracle Net8 Client 8.1.6■ Oracle Adapter for Named Pipes, SPX, or TCP/IP <p>The Net8 products are included on the distribution CD-ROM.</p> <p>Net8 or Net8 Client and the Oracle Adapter must be installed on the machine where the Oracle database server is installed. Net8 Server and the Oracle Adapter must be installed on the machine where the gateway is installed.</p>
Sybase	<p>Sybase Server or Client, Version 11.03, 11.5.1 or 11.9.2 is required. If Sybase Server is not on the same machine as the gateway, then Sybase Open client library Version 11.03 or 11.1.1 is required.</p>

Tested Configurations

The following table provides the tested configurations at Oracle, at the time of this document release. Oracle continues to provide support for the most recent releases of Oracle and non-Oracle systems in a timely manner. For current supported configuration information, please contact Oracle Support Services or visit <http://www.oracle.com/gateways>.

Gateway Configurations for the Oracle database server release 8.1.6

Database	Gateway and Operating System
Sybase Version 11.03	tg4sybs release 8.1.6 running on Sun SPARC Solaris Version 2.6 and 2.7
Sybase Version 11.5.1	tg4sybs release 8.1.6 running on Sun SPARC Solaris Version 2.6 and 2.7
Sybase Version 11.9.2	tg4sybs release 8.1.6 running on Sun SPARC Solaris Version 2.6 and 2.7

Database Compatibility Issues for Sybase

Sybase and Oracle databases function differently in some areas, causing compatibility problems. The following compatibility issues are described in this section:

- [Chained Mode](#)
- [Column Definitions](#)
- [Naming Rules](#)
- [Data Types](#)
- [Queries](#)
- [Locking](#)

Chained Mode

The gateway supports the ANSI-standard chained mode. Sybase stored procedures must be written for this mode. Running in chained mode allows the gateway to extend the Oracle two-phase commit protection to transactions updating Oracle and Sybase databases.

Column Definitions

By default, a Sybase table column cannot contain null values unless NULL is specified in the column definition. In compliance with the ANSI standard, the Sybase database option "allow nulls by default" can be set to true to change the default column definition to NULL.

For an Oracle table, null values are allowed in a column unless NOT NULL is specified in the column definition.

Naming Rules

Naming rule issues include the following:

- [Rules for Naming Objects](#)
- [Case Sensitivity](#)

Rules for Naming Objects

Oracle and Sybase use different database object naming rules. For example, the maximum number of characters allowed for each object name can be different. Also, the use of single and double quote marks, case sensitivity, and the use of alphanumeric characters can all be different.

See Also: *Oracle8i Reference* and Sybase documentation.

Case Sensitivity

The Oracle database server defaults to uppercase unless you surround identifiers with double quote characters. For example, to refer to a Sybase table called emp, enter the name with double quote characters, as follows:

```
SQL> SELECT * FROM "emp"@SYBS;
```

However, to refer to a Sybase table called emp owned by Scott from an Oracle application, enter the following:

```
SQL> SELECT * FROM "Scott"."emp"@SYBS;
```

If the Sybase table called emp is owned by SCOTT, a table owner name in uppercase letters, you can enter the owner name without double-quote characters, as follows:

```
SQL> SELECT * FROM SCOTT."emp"@SYBS;
```

or

```
SQL> SELECT * FROM scott."emp"@SYBS;
```

Oracle Corporation recommends that you surround all Sybase object names with double-quote characters and use the exact letter case for the object names as they appear in the Sybase data dictionary. This convention is not required when referring to the supported Oracle data dictionary tables or views listed in [Appendix B, "Data Dictionary"](#).

If existing applications cannot be changed according to these conventions, create views in Oracle to associate Sybase names to the correct letter case. For example, to

refer to the Sybase table emp from an existing Oracle application by using only uppercase names, define the following view:

```
SQL> CREATE VIEW EMP (EMPNO, ENAME, SAL, HIREDATE)
      AS SELECT "empno", "ename", "sal", "hiredate"
      FROM "emp"@SYBS;
```

With this view, the application can issue statements such as the following:

```
SQL> SELECT EMPNO, ENAME FROM EMP;
```

Using views is a workaround solution that duplicates data dictionary information originating in the Sybase data dictionary. You must be prepared to update the Oracle view definitions whenever the data definitions for the corresponding tables are changed in the Sybase database.

Data Types

Data type issues include the following:

- [Binary Literal Notation](#)
- [Data Type Conversion](#)

Binary Literal Notation

Oracle SQL uses hexadecimal digits surrounded by single quotes to express literal values being compared or inserted into columns defined as data type RAW.

This notation is not converted to syntax compatible with the Sybase VARBINARY and BINARY data types (a 0x followed by hexadecimal digits surrounded by single quotes).

For example, the following statement is not supported:

```
SQL> INSERT INTO BINARY_TAB@SYBS VALUES ('0xff')
```

where BINARY_TAB contains a column of data type VARBINARY or BINARY. Use bind variables when inserting into or updating VARBINARY and BINARY data types.

Data Type Conversion

Sybase does not support implicit date conversions. Such conversions must be explicit.

For example, the gateway issues an error for the following SELECT statement:

```
SELECT DATE_COL FROM TEST@SYBS WHERE DATE_COL = "1-JAN-1998";
```

To avoid problems with implicit conversions, add explicit conversions, as in the following:

```
SELECT DATE_COL FROM TEST@SYBS WHERE DATE_COL = TO_DATE("1-JAN-1998")
```

See Also: ["Data Type Conversion"](#) on page 5-4 for more information about restrictions on data types.

Queries

Query issues include the following:

- [Row Selection](#)
- [Empty Strings](#)
- [Empty Bind Variables](#)

Row Selection

Sybase evaluates a query condition for all selected rows before returning any of the rows. If there is an error in the evaluation process for one or more rows, no rows are returned even though the remaining rows satisfy the condition.

Oracle evaluates the query condition row-by-row and returns a row when the evaluation is successful. Rows are returned until a row fails the evaluation.

Empty Strings

Oracle processes an empty string in a SQL statement as a null value. Sybase processes an empty string as a single space. For example, the following statement:

```
SELECT "ename", "empno", "job" FROM "emp"@SYBS  
WHERE "ename" = ' ';
```

is processed by Sybase as follows:

```
SELECT ename, empno, job FROM emp WHERE ename = ' '
```

The gateway passes the empty string to the Sybase database without any conversion. If you intended an empty string to represent a null value, Sybase does not process the statement that way; it uses a space value. Avoid this problem by using NULL or IS NULL in the SQL statement instead of the empty string syntax, as in the following example:

```
SELECT "ename", "empno", "job" FROM "emp"@SYBS
WHERE "ename" IS NULL;
```

The gateway returns an empty string to the Oracle database server as a NULL value. This applies only to columns defined with a VARCHAR data type.

Empty Bind Variables

The gateway passes empty bind variables to the Sybase database as a NULL value. This applies only to columns defined with a VARCHAR data type.

Locking

The locking model for a Sybase database differs significantly from the Oracle model. The gateway depends on the underlying Sybase behavior, so Oracle applications that access Sybase through the gateway can be affected by the following possible scenarios:

- Read access might block write access
- Write access might block read access
- Statement-level read consistency is not guaranteed

See Also: Sybase documentation for information about the Sybase locking model.

Known Restrictions

Oracle Transparent Gateway for Sybase, Version 8.1.6, is architecturally different from the earlier versions of the gateway. If you encounter incompatibility problems not listed in this section or in “Known Problems” on page 1-14, please contact Oracle Support Services. The following section describes the known restrictions and includes suggestions for dealing with them when possible:

- [Transactional Integrity](#)
- [Transaction Capability](#)

- [COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors](#)
- [Stored Procedures](#)
- [Pass-Through Feature](#)
- [Sybase NCHAR and NVARCHAR Data Types](#)
- [SQL Syntax](#)
- [SQL*Plus COPY Command with Lowercase Table Names](#)
- [Database Links](#)
- [National Language Support](#)

The following restriction also applies:

- When negative numbers are used as the second parameter in a SUBSTR function, incorrect results are returned. This is due to incompatibility between the Oracle SUBSTR function and the equivalent in Sybase.

Note: If you have any questions or concerns about the restrictions, contact Oracle Support Services.

Transactional Integrity

The gateway cannot guarantee transactional integrity in the following cases:

- When a statement that is processed by the gateway causes an implicit commit in the target database
- When the target database is configured to work in autocommit mode

Note: Oracle corporation strongly recommends the following:

- If you know that executing a particular statement causes an implicit commit in the target database, then ensure that this statement is executed in its own transaction.
 - Do not configure the target database to work in autocommit mode.
-
-

Transaction Capability

The gateway does not support savepoints. If a distributed update transaction is under way involving the gateway and a user attempts to create a savepoint, the following error occurs:

ORA-02070: database *dblink* does not support savepoint in this context

By default, the gateway is configured as COMMIT_CONFIRM and it is always the commit point site when the Sybase database is updated by the transaction.

See Also: *Oracle8i Distributed Database Systems* for more information and restrictions of the COMMIT_CONFIRM mode.

COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors

Any COMMIT or ROLLBACK issued in a PL/SQL cursor loop closes all open cursors, which can result in the following error:

ORA-1002: fetch out of sequence

To prevent this error, move the COMMIT or ROLLBACK statement outside the cursor loop.

Stored Procedures

Changes issued through stored procedures that embed commits or rollbacks cannot be controlled by the Oracle transaction manager or Oracle COMMIT or ROLLBACK commands.

Stored procedures that return a result set can be executed, but the result set is skipped. Any output parameters and return values are returned.

Pass-Through Feature

DDL statements executed by Sybase using the gateway pass-through feature might fail if they are in a multi-statement transaction. Set the Sybase option "ddl in tran" to allow DDL statements in a transaction.

Oracle Corporation recommends that you place a DDL statement in its own transaction when executing such a statement with the pass-through feature. An explicit COMMIT must be issued after the DDL statement.

If the SQL statements being passed through the gateway result in an implicit commit at the Sybase database, the Oracle transaction manager is unaware of the

commit and an Oracle ROLLBACK command cannot be used to roll back the transaction.

Sybase NCHAR and NVARCHAR Data Types

The gateway cannot select a column defined with a Sybase NCHAR or NVARCHAR data type.

SQL Syntax

This section lists restrictions on the following SQL syntax:

- [SELECT FOR UPDATE Statement](#)
- [WHERE CURRENT OF Clause](#)
- [CONNECT BY Clause](#)
- [Subqueries in UPDATE Statement](#)
- [ROWID](#)
- [Subqueries in INSERT Statement](#)

See Also: [Appendix A, "Supported SQL Syntax and Functions"](#)
for more information about restrictions on SQL syntax.

SELECT FOR UPDATE Statement

The SELECT FOR UPDATE statement is not supported.

WHERE CURRENT OF Clause

UPDATE and DELETE statements with the WHERE CURRENT OF clause are not supported by the gateway because they rely on the Oracle ROWID implementation. To update or delete a specific row through the gateway, a condition style WHERE clause must be used.

CONNECT BY Clause

The gateway does not support the CONNECT BY clause in a SELECT statement.

Subqueries in UPDATE Statement

Subqueries in the SET clause of an UPDATE statement are not supported.

Subqueries can be specified in the WHERE clause of an UPDATE statement. Each subquery, however, must reference a Sybase table. For example, using the table GTW_EMP, the following statement results in a 10% salary increase for all employees working in the RESEARCH department:

```
SQL> UPDATE "GTW_EMP"@SYBS SET "SAL"="SAL" * 1.1
      2 WHERE "DEPTNO"=(SELECT "DEPTNO" FROM "GTW_DEPT"@SYBS
      3 WHERE "DNAME"='RESEARCH');
```

If "GTW_DEPT"@SYBS is replaced by "DEPT" in the subquery where DEPT is the same table but located in the Oracle database, the following error results after the statement is issued:

ORA-02025: All tables in the SQL statement must be at the remote database.

ROWID

The Oracle ROWID implementation is not supported.

Subqueries in INSERT Statement

Subqueries of INSERT statements cannot use multiple aliases for the same table. For example, the following statement is not supported:

```
SQL> INSERT INTO "emp_target"@SYBS
      SELECT a."empno" FROM "emp_source"@SYBS a,
      "emp_source"@SYBS b WHERE b."empno"=9999
```

SQL*Plus COPY Command with Lowercase Table Names

The gateway does not support the SQL*Plus COPY command for lowercase table names.

Database Links

The gateway is not multithreaded and cannot support shared database links. Each gateway session spawns a separate gateway process and connections cannot be shared.

National Language Support

The gateway supports all single byte ASCII character sets and the following multibyte character sets:

- JA16EBCDIC930
- JA16EUC
- JA16EUCYEN
- JA16SJIS
- JA16SJISYEN
- JA16VMS

See Also: *Oracle8i Distributed Database Systems* and *Oracle8i National Language Support Guide*.

Known Problems

This section describes known problems and includes suggestions for correcting them when possible. If you have any questions or concerns about the problems, contact Oracle Support Services. A current list of problems is available online. Contact your local Oracle Corporation office for information about accessing the list.

The following known problems are described in this section:

- [FLOAT Data Type](#)
- [VARBINARY Data Type](#)
- [Encrypted Format Login](#)
- [Aggregate Function with CREATE TABLE or CREATE VIEW Statement](#)
- [Date Arithmetic](#)
- [Sybase IMAGE and TEXT Data Types](#)
- [String Functions](#)
- [Schema Names and PL/SQL](#)
- [Data Dictionary Views and PL/SQL](#)
- [Stored Procedures](#)

FLOAT Data Type

The FLOAT data type precision is (7).

VARBINARY Data Type

The VARBINARY data type is reported as BINARY.

Encrypted Format Login

The Oracle database server supports an Oracle initialization parameter, DBLINK_ENCRYPT_LOGIN. When this parameter is set to TRUE, the password for the login user ID is not sent over the network.

If this parameter is set to TRUE in the initialization parameter file used by the Oracle database server, you must change the setting to FALSE to allow Oracle to communicate with the gateway.

Aggregate Function with CREATE TABLE or CREATE VIEW Statement

The Oracle database server does not send the gateway a SELECT statement containing an aggregate function that is part of a CREATE TABLE or CREATE VIEW statement. For example, it does not send the following statement:

```
CREATE TABLE sum_calls_table AS
  SELECT SUM(calls_abandoned), SUM(calls_completed),
  SUM(calls_failed) FROM monthly_calls@SYBS;
```

Instead, Oracle interprets what the SQL statement requests and sends the gateway a statement or statements to retrieve the data required for the request. After the data is retrieved, Oracle performs the aggregate function originally requested and passes the results to the application.

A solution to this problem is to use a different series of SQL statements. For example, instead of using the CREATE TABLE statement in the above example, use the following statements:

```
DROP TABLE sum_calls_table;
CREATE TABLE sum_calls_table (x1sum NUMBER, x2sum NUMBER,
  x3sum NUMBER);
DECLARE
  x1 NUMBER;
  x2 NUMBER;
  x3 NUMBER;
BEGIN
```

```
SELECT SUM(calls_abandoned), SUM(calls_completed),  
SUM(calls_failed) INTO x1, x2, x3 FROM monthly_calls@SYBS;  
INSERT INTO sum_calls_table VALUES (x1, x2, x3);  
END;  
/
```

Date Arithmetic

The following SQL expressions do not function correctly with the gateway:

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

Statements with the above expressions are sent to the Sybase database without any translation. Since Sybase does not support these date arithmetic functions, the statements return an error.

Sybase IMAGE and TEXT Data Types

The following restrictions apply when using IMAGE and TEXT data types:

- An unsupported SQL function cannot be used in a SQL statement that accesses a column defined as Sybase data type IMAGE or TEXT.
- You cannot use SQL*Plus to select data from a column defined as Sybase data type IMAGE or TEXT when the data is greater than 80 characters in length. Oracle Corporation recommends using Pro*C or Oracle Call Interface to access such data in a Sybase database.
- IMAGE and TEXT data types must be NULLABLE for INSERT or UPDATE to work.
- A table including a IMAGE or TEXT column must have a unique index defined on the table or the table must have a separate column that serves as a primary key.
- IMAGE and TEXT data cannot be read through pass-through queries.

The gateway does not support the PL/SQL function COLUMN_VALUE_LONG of the DBMS_SQL package.

See Also: [Appendix A, "Supported SQL Syntax and Functions"](#).

String Functions

If you concatenate numeric literals using the " || " or CONCAT operator when using the gateway to query a Sybase database, the result is an arithmetic addition. For example, the result of the following statement is 18:

```
SQL> SELECT 9 || 9 FROM DUAL@SYBS;
```

The result is 99 when using Oracle to query an Oracle database.

Schema Names and PL/SQL

If you do not prefix a Sybase database object with its schema name in a SQL statement within a PL/SQL block, the following error message occurs:

```
ORA-6550 PLS-201 Identifier table_name must be declared.
```

Change the SQL statement to include the schema name of the object.

Data Dictionary Views and PL/SQL

You cannot refer to data dictionary views in SQL statements that are inside a PL/SQL block.

Stored Procedures

Stored procedures with output parameters defined with a CHAR data type return output parameters with VARCHAR data types.

Stored procedures with input or output parameters defined with NUMERIC or DECIMAL data types that are passed with either a NULL value or no value for input will return output values without the fractional part. To prevent this from happening, always specify an input parameter value, even if it is not used.

Introduction

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

This chapter contains the following sections:

- [Oracle Heterogeneous Services](#)
- [Oracle Transparent Gateway for Sybase](#)
- [Gateway Architecture](#)
- [Gateway Process Flow](#)
- [Additional Features](#)

Oracle Heterogeneous Services

The generic core of the Oracle Transparent Gateway for Sybase is incorporated into the database as the Heterogeneous Services facility. Heterogeneous Services (HS) extend the Oracle SQL engine to recognize the SQL and procedural capabilities of the remote non-Oracle system and the mappings required to obtain necessary data dictionary information. Additionally, since Heterogeneous Services is integrated with the Oracle database server, the transaction coordination of the server maintains the transaction coordination between Oracle and the remote non-Oracle system. For example, Heterogeneous Services provide the two-phase commit protocol to ensure distributed transaction integrity, even for non-Oracle systems that do not natively support two-phase commit.

Oracle Transparent Gateway for Sybase, together with the Heterogeneous Services component that is integrated in the Oracle8i server, provide information for and connectivity to Sybase.

See Also: *Oracle8i Distributed Database Systems.*

Oracle Transparent Gateway for Sybase

Using Oracle SQL, Oracle client applications can access Sybase data as if the data were stored in an Oracle table. Data residing in Oracle and Sybase databases can be accessed by a single SQL statement, performing heterogeneous joins and subselects. This means you can develop one set of portable applications to use against Oracle and Sybase databases. You can continue to develop new information systems without losing your investment in existing data and applications.

Transactions updating Oracle and Sybase databases are automatically protected by the Oracle two-phase commit feature. Use of synonyms is another Oracle feature. By setting up synonyms in the Oracle database server that point to database links to Sybase tables, the physical location of the data is transparent to the client application. This allows future migration of data from Sybase to Oracle to be transparent to the client applications.

The gateway requires only the Oracle database server and Net8. All other Oracle products are optional. However, using other Oracle products with the gateway can greatly extend the gateway's capabilities.

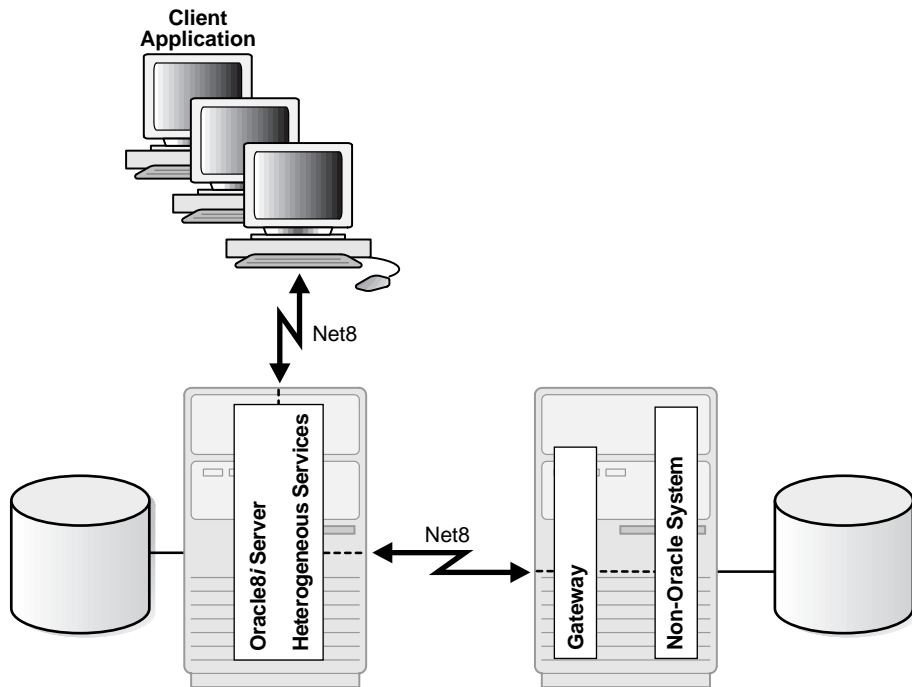
Gateway Architecture

The gateway is invoked by the listener. The gateway is not multi-threaded and cannot support shared database links. Each gateway session spawns a separate gateway process, and connections cannot be shared.

The gateway resides on the Sun SPARC Solaris machine with the Sybase database or Sybase client (in which case the Sybase database can reside on a separate machine). The Oracle database server can reside on the same machine as the gateway or on another machine.

Note: The non-Oracle system shown in [Figure 2-1](#) and [Figure 2-2](#) represents Sybase.

The gateway interacts with the Oracle database server to interface between client applications and Sybase, as shown in [Figure 2-1](#) on page 2-4.

Figure 2–1 Gateway Processing

- Client applications, such as Developer, connect to the Oracle database server by using Net8.
- The Oracle database server, which includes Heterogeneous Services and the database itself, resides on a single system. This Oracle database server also stores definitions of database links for Sybase.
- The gateway to Sybase and the Sybase database server itself, which includes the Sybase database, reside on a second system. The Oracle database server on the first system uses Net8 to connect directly to the Sybase gateway on the second system.
- On the second system, the Sybase gateway accesses the Sybase database server by using whatever networking software the target supports.

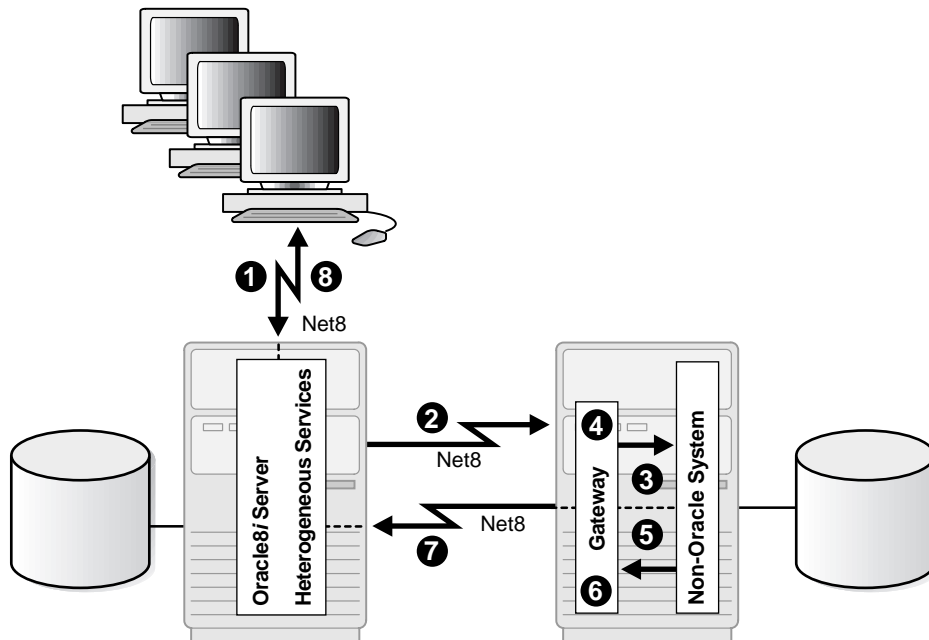
The Oracle database server and the gateway work together to present the appearance of a single Oracle database to the client. All data accessed by the client

appears to reside in a single Oracle database. The client application sends a request to the Oracle database server, and the Oracle database server sends the request to the gateway. For the first transaction in a session, the gateway logs into Sybase using a username and password that is valid in the Sybase database. The gateway converts the SQL statement to a native Sybase statement, and Sybase performs the request. The gateway converts the retrieved data to a format compatible with the Oracle database server and returns the results to the Oracle database server, which returns the results to the client application.

Gateway Process Flow

Figure 2-2 shows a typical gateway process flow. The steps explain the sequence of events that occurs when a client application queries the Sybase database through the gateway.

Figure 2-2 Gateway Process Flow



1. The client application sends a query over Net8 to the Oracle database server.

2. The Oracle database server sends the query over to the gateway, again using Net8.
3. For the first transaction in a session, the gateway logs into Sybase using a username and password that is valid in the Sybase database.
4. The gateway converts the Oracle SQL statement into an SQL statement understood by Sybase.
5. The gateway retrieves data using Sybase SQL statements.
6. The gateway converts retrieved data into a format compatible with the Oracle database server.
7. The gateway returns query results to the Oracle database server, again using Net8.
8. The Oracle database server passes the query results to the client application by using Net8. The database link remains open until the gateway session is finished or the database link is explicitly closed.

Additional Features

This section describes the following additional features provided by the Oracle Transparent Gateway for Sybase:

- [Remote Data Access](#)
- [Elimination of Unnecessary Data Duplication](#)
- [Heterogeneous Database Integration](#)
- [Application Development and End User Tools](#)
- [Two-Phase Commit and Multi-Site Transactions](#)
- [Query Optimization](#)
- [Error Mapping and Logging](#)
- [Stored Procedure Execution](#)
- [Pass-Through Feature](#)

Remote Data Access

Applications can take advantage of Oracle client-server capability to connect to a remote server using Net8. The server can then connect to the gateway using a database link. You have more flexibility in locating your data because the Oracle architecture enables network connections between each of the components.

With remote access, you can move application development onto cost-efficient workstations or microcomputers. Without remote access, you are limited to the data available in the local environment. With remote access, your data sources are virtually unlimited. Remote access also enables you to choose the best environment for your users. For example, data might be located on a platform that supports only character-mode interfaces, but users can access the data from desktop platforms that support graphical user interfaces.

Elimination of Unnecessary Data Duplication

The gateway gives applications direct access to Sybase data, so the need for uploading and downloading large quantities of data to other locations is eliminated. You instead access the data where it is, when you want it, without having to move the data between machines and risk unsynchronized and inconsistent data. Avoiding data duplication reduces the disk storage requirements over all your systems.

Heterogeneous Database Integration

The Oracle database server can accept a SQL statement that queries data stored in several different databases. The Oracle database server with heterogeneous services processes the SQL statement and passes the appropriate SQL directly to other Oracle databases and through gateways to non-Oracle databases. The Oracle database server then combines the results and returns them to the client. This enables a query to be processed that spans Sybase, other databases, and local and remote Oracle data.

Application Development and End User Tools

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

Oracle and Sybase data with one set of tools. These tools can run on remote machines connected through Net8 to the Oracle database server.

Two-Phase Commit and Multi-Site Transactions

In a distributed database system, the network might fail during a distributed transaction. The Oracle transaction model uses a two-phase commit protocol to protect the databases during the period of committing data at sites participating in a distributed transaction. This feature ensures that all database servers participating in the transaction commit or roll back the transaction statements. The gateway supports this two-phase commit protocol. Only one Sybase database is allowed for each update transaction.

Query Optimization

Whenever possible, the Oracle database server passes the entire query to the non-Oracle system to utilize the indexes and statistics of the non-Oracle system tables.

When a query that involves multiple databases is processed, the Oracle database server passes optimized statements to the remote servers and gateways involved in the query to minimize the amount of data returned across the network.

See Also: *Oracle8i Designing and Tuning for Performance* and *Oracle8i Distributed Database Systems* for information about collocated inline views and joins.

Error Mapping and Logging

The gateway provides error mapping. It maps the Sybase error to an Oracle database server error message and adds all of the relevant error messages generated by Sybase. You can route messages to the client application, an operator console, an error log, or any combination of these destinations as needed. Error mapping provides database transparency for applications.

Stored Procedure Execution

Using the procedural feature, the gateway can execute stored procedures defined in Sybase databases. There is no requirement to relink the gateway or define the procedure to the gateway, but the procedure's access privileges must permit access by the gateway.

See Also: [Executing Stored Procedures and Functions](#) on page 5-6.

Pass-Through Feature

Commands and statements specific to the Sybase database can be passed through the gateway for execution by Sybase.

See Also: [Using the Pass-Through Feature](#) on page 5-5.

Installing the Gateway

This chapter explains how to install the gateway software from the distribution CD-ROM. This chapter includes the following sections:

- [Before You Begin](#)
- [Installing the Gateway from the CD-ROM](#)
- [De-installing the Gateway](#)

Before You Begin

Confirm that you have met all of the hardware and software requirements described in Chapter 1. You should also determine the configuration of the gateway components on the system, identify the machines on which the Oracle database server, the gateway, and the Sybase database reside, and verify the administration roles. Refer to the following sections:

- [Determine the Configuration of the Gateway Components](#)
- [Installation Worksheet](#)

Determine the Configuration of the Gateway Components

[Figure 3–1, "Gateway Component Configuration"](#), is a system diagram that gives an example of one possible system configuration. To sketch your own system, first draw a box. This represents a single system. Next, for each component you wish to install on the system, draw a box inside the box representing your system. Label these boxes with their respective component names. Follow the same procedure to represent a second system.

Draw a line with an arrow on both ends between the two systems to represent communication between them. Draw a box or boxes to represent client applications, and draw a line between them and the Oracle database server to represent communications between the client applications and the Oracle database server. Label the machines on which the Oracle database server, the gateway, and the non-Oracle database reside.

Identify the machines on which Sybase and the Oracle database server reside and where the gateway will be installed. Write this information on the [Installation Worksheet](#). After you determine the configuration and identify the machines, verify that the machines and the software meet the hardware and software requirements described in Chapter 1. The supported versions of Sun SPARC Solaris, Sybase database and communication software, and the Oracle database server must all be installed.

Important: Verify that you have Oracle database administration access and privileges on the machines on which the gateway and the Oracle database server are to be installed.

Figure 3–1 Gateway Component Configuration

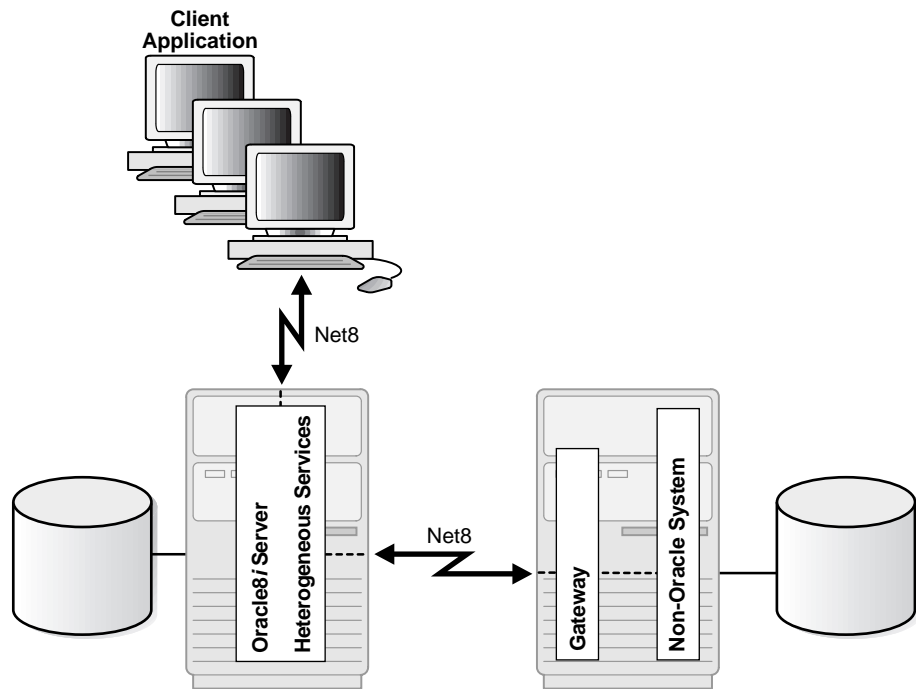


Figure 3–1, "Gateway Component Configuration", is a system diagram that gives an example of one possible system configuration. In this diagram, the Oracle8i database server, which includes Heterogeneous Services and the database files, resides on a single machine. Client applications access the server over Net8. The gateway and its respective non-Oracle system and database reside on a second machine. The two systems communicate over Net8 so that the Oracle database server accesses the non-Oracle system through the gateway.

Installation Worksheet

Fill in the values on the following Installation Worksheet, so you can refer to it during the configuration process.

Table 3–1 Installation Worksheet

Description	Value
Oracle database server machine name	
Oracle database server platform (OS and its version number)	
\$ORACLE_HOME of Oracle database server (full path name)	
Gateway machine name	
Gateway machine platform (OS and its version number)	
\$ORACLE_HOME of the gateway (full path name)	
Sybase client library location (full path name)	
Sybase Open Client libraries location or Sybase Adaptive Server location on the gateway machine	
Version of the Sybase Server to which the gateway will connect	
Name of the Sybase Adaptive Server to which the gateway will connect	
Name of the Sybase database to which the gateway will connect	

Installing the Gateway from the CD-ROM

Install the Gateway software from the CD-ROM as follows:

1. Log on to Sun SPARC Solaris as the Oracle Database Administrator (DBA) user.

If you are not currently a DBA user, contact the system administrator to create a database administrator login user ID.

See Also: The Preinstallation chapter of the *Oracle8i Installation Guide Release 2 (Oracle v8.1.6) for Sun SPARC Solaris*.

2. Create the \$ORACLE_HOME directory.

Create the \$ORACLE_HOME directory where the gateway will be installed and write the value on the [Installation Worksheet](#). The gateway installation should be under a new \$ORACLE_HOME. Do not install under an existing \$ORACLE_HOME that already has either the Oracle database server or one or more other Oracle products installed.

When you create the directory, use the version number as part of the path name to allow different versions of the gateway to be installed under one Oracle directory tree. For example:

```
$ mkdir /oracle
$ mkdir /oracle/gateway
$ mkdir /oracle/gateway/8.1.6
$ chown oracle /oracle/gateway/8.1.6
$ chgrp dba /oracle/gateway/8.1.6
$ chmod 755 /oracle/gateway/8.1.6
```

3. Set the environment variables path.

Set the \$ORACLE_HOME environment variable path to point to the directory you created in step 2 above. The command you use to set this variable depends upon which UNIX shell you are using. For example, if you are using either the Bourne Shell or Korn Shell, enter the following:

```
$ ORACLE_HOME=/oracle/gateway/8.1.6; export ORACLE_HOME
```

If you are using the C Shell, enter the following:

```
$ setenv ORACLE_HOME /oracle/gateway/8.1.6
```

Set the ORACLE_TERM environment variable to point to the terminal type. This command depends on which UNIX shell you are using. For example, if you are using either the Bourne Shell or Korn Shell, enter the following:

```
$ ORACLE_TERM=vt100; export ORACLE_TERM
```

If you are using the C Shell, enter the following:

```
$ setenv ORACLE_TERM vt100
```

4. Set up and start the Oracle Universal Installer.

Insert the CD-ROM into the CD-ROM drive, change directories to this drive, and start the Oracle Universal Installer. Enter the following commands:

```
$ cd /cdrom/cdrom0
$ ./runInstaller
```

See Also: *Oracle 8i Installation Guide Release 2 (8.1.6) for Sun SPARC Solaris* for general information about installing Oracle products and using the Oracle Universal Installer.

5. Run the Oracle Universal Installer and install the gateway.

The Oracle Universal Installer screens and the sequence in which they appear depend on the platform. Default values are selected for you and should be used unless you have reason to change them. Refer to [Table 3–2, "Installation Prompts and Responses"](#), as a guide to the installation. Follow the instructions in the Response column for each entry in the Screen column.

Table 3–2 Installation Prompts and Responses

Screen	Response
Full path of the products.jar file which contains the product or products you want to install.	The Oracle Universal Installer extracts and displays this value from its configuration file. Do not change the text in the Source field.
Full path of the gateway \$ORACLE_HOME	Enter the \$ORACLE_HOME directory path in which the gateway will be installed. You can set the default location of the \$ORACLE_HOME environment variable before you start the Oracle Universal Installer.
Installation Types	Custom – If you selected this installation type, then you can optionally install the Oracle Universal Installer on your machine in addition to the Sybase gateway. Also, you can select the languages in which the Sybase gateway runs. Typical – If you selected this installation type, both the Sybase gateway and the Oracle Universal Installer are installed on your machine. The default language selected during install is English.

Table 3–2 (Cont.) Installation Prompts and Responses

Screen	Response
Sybase Server Version	<p>Select the Sybase version to which the gateway will connect. The supported versions are: 11.03, 11.5.1/11.9.2, or both. If you select:</p> <p>11.0.3 – The installation proceeds without any further input pertaining to the OC library version.</p> <p>11.5.1/11.9.2 – The next screen prompts you with a message stating that the gateway cannot be used to a Sybase Server version 11.03.</p> <p>both – The next screen prompts you with a message stating that the Sybase interfaces file entry for Sybase Server versions 11.5.1/11.9.2 should contain the TLI entry.</p>
Full path to where Sybase is installed.	Enter the full directory path to where the Sybase Open Client libraries or the Sybase Server is installed. The default location is the value of the SYBASE environment variable if you had set its value prior to starting the installer.
The name of the Sybase Server to which the gateway will connect.	Enter the Sybase Server entry in the interfaces file to which the gateway will connect. The default name is the value of the DSQUERY environment variable if you had set its value prior to starting the installer.
The name of the Sybase database to which the gateway will connect.	Enter the name of the database to which the gateway will connect.
Net8 Configuration Assistant	Check the box "Perform typical configuration."

6. Verify whether the software was successfully installed.

You can verify that the installation was successful after the Oracle Universal Installer confirms that the installation has been completed. Check the contents of the installation log file located in the following file, where `orainventory_location` is the directory in which OraInventory is located:

```
orainventory_location/logs/installActions.log
```

De-installing the Gateway

De-install the gateway as follows:

1. Start the Oracle Universal Installer: see Step 4, "[Set up and start the Oracle Universal Installer.](#)"
2. Click the [De-install Products] button on the Welcome dialog box or the [Installed Products ...] button available on any Oracle Universal Installer window. The Inventory dialog box appears and displays a list of installed products.
3. Click to select the gateway product in the Inventory dialog box, and click the Remove button.

Only the files that were copied into the \$ORACLE_HOME directory during the gateway installation are removed when the gateway is de-installed. You must manually remove all other related files.

Configuring the Gateway

After installing the gateway, perform the following tasks to configure the gateway for Sybase:

- [Configuring the Gateway](#)
- [Configuring Net8 Listener for the Gateway](#)
- [Configuring the Oracle Database Server for Gateway Access](#)
- [Performing Optional Configuration Tasks](#)
- [Creating Database Links](#)
- [Configuring the Gateway for Multiple Sybase Databases](#)

Configuring the Gateway

Perform the following tasks to configure the Oracle Transparent Gateway for Sybase.

Task 1: Choose a System ID for the Gateway

The gateway system identifier (SID) is an alphanumeric character string that identifies a gateway instance. You need one gateway instance, and therefore one gateway SID, for each Sybase database you are accessing. The SID is used as part of the file name for the initialization parameter file. The default SID is `tg4sybs`.

You can define a gateway SID, but using the default of `tg4sybs` is easier because you do not need to change the initialization parameter file name. However, if you want to access two Sybase databases, you need two gateway SIDs, one for each instance of the gateway. If you have one Sybase database and want to access it sometimes with one set of gateway parameter settings, and other times with different gateway parameter settings, you can do that by having multiple gateway SIDs for the single Sybase database.

Task 2: Customize the Initialization Parameter File

The initialization parameter file must be available when the gateway is started. During installation, the following default initialization parameter file is created:

```
$ORACLE_HOME/tg4sybs/admin/inittg4sybs.ora
```

where `$ORACLE_HOME` is the directory under which the gateway is installed.

If you are not using `tg4sybs` as the gateway SID, you must rename the initialization parameter file using the SID you chose in Task 1. This default initialization parameter file is sufficient for starting the gateway, verifying a successful installation, and running the demonstration scripts.

In the initialization parameter file, specify the Sybase connection as follows:

```
HS_FDS_CONNECT_INFO=server_name.database_name
```

The entries for both the *server_name* and *database_name* are case-sensitive.

Additionally, set the Sybase environment variable, as follows:

```
Set SYBASE=sybase_dir
```

sybase_dir is the directory where the Sybase Adaptive Server is installed or the directory where the Sybase Open Client libraries are installed when the Sybase

Adaptive Server is on a different machine from the gateway. This directory must be the same as that specified for the LD_LIBRARY_PATH.

A number of initialization parameters can be used to modify gateway behavior. You might want to change the initialization parameter file later to meet system requirements.

See Also: [Appendix C, "Heterogeneous Services Initialization Parameters"](#) and the *Oracle8i Distributed Database Systems* for more information about customizing the initialization parameter file.

Configuring Net8 Listener for the Gateway

The gateway requires Net8 to provide transparent data access. After configuring the gateway, configure Net8 to work with the gateway.

Task 1: Configure Net8 TNS Listener for the Gateway

Net8 uses the TNS listener to receive incoming connections from a Net8 client. The TNS listener and the gateway must reside on the same machine.

The TNS listener listens for incoming requests from the Oracle database server. For the TNS listener to listen for the gateway, information about the gateway must be added to the TNS listener configuration file, `listener.ora`. This file is located in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the `$ORACLE_HOME` directory under which the TNS listener is installed. This is the same `$ORACLE_HOME` directory under which the gateway is installed.

Note: If Net8 is reinstalled, the original `listener.ora` file is renamed and a new `listener.ora` file is put into the `$ORACLE_HOME/network/admin` directory.

The following entries must be added to the `listener.ora` file:

- A list of Net8 addresses on which the TNS listener listens
- The gateway that the TNS listener starts in response to incoming connection requests

Example of Address to Listen On in listener.ora File

The Oracle database server accesses the gateway using Net8 and the TCP/IP protocol adapter. The following is the syntax of the connect descriptor entry in the `listener.ora` file:

```
LISTENER=
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=host_name)
    (PORT=port_number) )
```

where:

host_name is the name of the machine on which the gateway is installed.

port_number specifies the port number used by the TNS listener. If you have other listeners running on *host_name*, the value of *port_number* must be different from the other listeners' port numbers.

Example of Gateway to Start in listener.ora File

To direct the TNS listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file with the following syntax:

```
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=gateway_sid)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4sybs)
    )
  )
```

where:

gateway_sid specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.

oracle_home_directory specifies the Oracle home directory where the gateway resides.

tg4sybs specifies the Oracle Transparent Gateway for Sybase.

If you are already running a TNS listener that listens on multiple database SIDs, add only the following syntax to SID_LIST in the existing `listener.ora` file:

```
SID_LIST_LISTENER=
(SID_LIST=
  (SID_DESC=.
    .
  )
  (SID_DESC=.
    .
  )
  (SID_DESC=
    (SID_NAME=gateway_sid)
    (ORACLE_HOME=oracle_home_directory)
    (PROGRAM=tg4sybs)
  )
)
```

See Also: *Net8 Administrator's Guide* for information about changing the `listener.ora` file.

Task 2: Stop and Start the TNS Listener for the Gateway

The TNS listener must be started to initiate the new settings, as follows:

1. Set the `PATH` environment variable to access the commands in the directory `$ORACLE_HOME/bin` where the gateway is installed. If you have the Bourne or Korn Shell, enter the following:

```
$ PATH=$ORACLE_HOME/bin:$PATH;export PATH
```

If you have the C Shell, enter the following:

```
$ setenv PATH $ORACLE_HOME/bin:$PATH
```

2. Set the `LD_LIBRARY_PATH` to include the Sybase libraries. If you have the Bourne or Korn Shell, enter the following:

```
$ LD_LIBRARY_PATH=$SYBASE/lib:$ORACLE_HOME/lib:$LD_LIBRARY_PATH;export LD_
LIBRARY_PATH
```

If you have the C Shell, enter the following:

```
$ setenv LD_LIBRARY_PATH $SYBASE/lib:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Note: SYBASE is a Sybase environment variable. See your Sybase documentation for information about setting the environment variable to access the Sybase database.

3. If the listener is already running, use the `lsnrctl` command to stop the listener and then start it with the new settings, as follows:

```
$ lsnrctl stop
$ lsnrctl start
```

4. Check the status of the listener with the new settings, as follows:

```
$ lsnrctl status
```

The following is an example of output from a `lsnrctl` status check:

```
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=204.179.99.15)(PORT=1551))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Solaris: Version 8.1.6.0.0 - Production
Start Date                05-DEC-1999 14:07:29
Uptime                    0 days 21 hr. 9 min. 25 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File   /parla/test40/sybs/network/admin/listener.ora
Listener Log File         /parla/test40/sybs/network/log/listener.log
Services Summary...
  tg4sybs                  has 1 service handler(s)
The command completed successfully
```

In this example, `tg4sybs` is the default SID value assigned during installation. You can use any valid ID for the SID, or keep the default.

Note: You must use the same SID value in the `tnsnames.ora` file, the `listener.ora` file.

Configuring the Oracle Database Server for Gateway Access

Any Oracle client connected to the Oracle database server can access Sybase data through the gateway. The Oracle client and the Oracle database server can reside on different machines. The gateway accepts connections only from the Oracle database server.

Before you use the gateway to access Sybase data you must configure the Oracle database server to enable communication with the gateway over Net8.

Configuring Oracle Net8 for the Oracle Database Server

To configure the server you add connect descriptors to the `tnsnames.ora` file. You cannot use the Net8 Assistant or the Net8 Easy Config tools to configure the `tnsnames.ora` file. You must edit the file manually.

See Also: *Oracle8i Administrator's Guide* for information about editing the `tnsnames.ora` file.

For the Oracle database server to access the gateway, it needs a service name entry or a connect descriptor name entry in the `tnsnames.ora` file to tell the Oracle database server where to make connections. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database server is installed. The `tnsnames.ora` file is required by the Oracle database server accessing the gateway, but not by the gateway.

Configuring `tnsnames.ora`

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is an example of the Net8 entries using TCP/IP protocol needed for the Oracle database server to access the gateway:

```
connect_descriptor=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (HOST=host_name)
      (PORT=port_number)
    )
    (CONNECT_DATA=
      (SID=gateway_sid)
    )
    (HS=OK) )
```

where:

connect_descriptor is the description of the object to connect to as specified when creating the database link, such as tg4sybs.

Check the `sqlnet.ora` file in the Oracle database server's `$ORACLE_HOME` for the following lines:

- `names.directory_path = (TNSNAMES, HOSTNAME)`
- `names.default_domain = world`
- `name.default_zone = world`

Note: If the Oracle database server is on Sun SPARC Solaris, the file is `$ORACLE_HOME/network/admin/sqlnet.ora`.

If the `sqlnet.ora` file has these lines, *connect_descriptor* must end with the extension `.world`.

TCP is the TCP protocol used for TCP/IP connections.

host_name specifies the machine where the gateway is running.

port_number matches the port number used by the Net8 TNS listener that is listening for the gateway. The TNS listener's port number can be found in the `listener.ora` file used by the TNS listener. See "Example of Address to Listen On in listener.ora File" on page 4-4.

gateway_sid specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the TNS listener that is listening for the gateway. See "Task 1: Configure Net8 TNS Listener for the Gateway" on page 4-3 for more information.

(HS=OK) specifies that this connect descriptor uses the Oracle Heterogeneous Services option.

Performing Optional Configuration Tasks

You can perform the following configuration tasks:

- [Configuring for Two-Phase Commit](#)
- [Specifying an Owner](#)

Configuring for Two-Phase Commit

The gateway supports the following transaction capabilities:

- COMMIT_CONFIRM
- READ_ONLY
- SINGLE_SITE

By default, the gateway runs in COMMIT_CONFIRM transaction mode. When the Sybase database is updated by a transaction, the gateway becomes the commit point site. The Oracle database server commits the unit of work in the Sybase database after verifying that all Oracle databases in the transaction have successfully prepared the transaction. Only one gateway can participate in an Oracle two-phase commit transaction as the commit point site.

See Also: *Oracle8i Distributed Database Systems* for information about the two-phase commit process.

To enable the COMMIT_CONFIRM transaction mode, create a recovery account and password and create a log table. The log table, called HS_TRANSACTION_LOG, is where two-phase commit transactions are recorded.

Task 1: Create a Recovery Account and Password

For the gateway to recover distributed transactions, a recovery account and password must be set up in the Sybase database. By default, both the user name of the account and the password are RECOVER. The name of the account can be changed with the gateway initialization parameter HS_FDS_RECOVERY_ACCOUNT. The account password can be changed with the gateway initialization parameter HS_FDS_RECOVERY_PWD.

Note: Oracle Corporation recommends that you use the default value RECOVER for the user name and password.

1. Set up a user account in the Sybase database. Both the user name and password must be a valid Sybase user name and password.
2. In the initialization parameter file, set the following gateway initialization parameters:
 - HS_FDS_RECOVERY_ACCOUNT to the user name of the Sybase user account you set up for recovery.
 - HS_FDS_RECOVERY_PWD to the password of the Sybase user account you set up for recovery.

See Also: For information about editing the initialization parameter file, see “Task 2: Customize the Initialization Parameter File” on page 4-2. For information about HS_FDS_RECOVERY_ACCOUNT and HS_FDS_RECOVERY_PWD, see [Appendix C, "Heterogeneous Services Initialization Parameters"](#).

Task 2: Create the Transaction Log Table

When configuring the gateway for two-phase commit, a table must be created in the Sybase database for logging transactions. The gateway uses the transaction log table to check the status of failed transactions that were started at the Sybase database by the gateway and registered in the table. Updates to the transaction log table cannot be part of an Oracle distributed transaction.

Note: The information in the transaction log table is required by the recovery process and must not be altered. The table must be used, accessed, or updated only by the gateway.

The table consists of two columns, GLOBAL_TRAN_ID, data type CHAR(64) NOT NULL and TRAN_COMMENT, data type CHAR(255).

Create a transaction log table in the user account you created in ["Task 1: Create a Recovery Account and Password"](#). Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database

where the Sybase update takes place. Also, the transaction log table must be created under the owner of the recovery account.

Note: To utilize the transaction log table, users of the gateway must be granted privileges on the table.

To create a transaction log table use the `tg4sybs_tx.sql` script, located in the directory `$ORACLE_HOME/tg4sybs/admin` where `$ORACLE_HOME` is the directory under which the gateway is installed. Use `isql` to execute the script, as follows:

```
$ isql -Urecovery_account -Precovery_account [-Sserver] -itg4sybs_tx.sql
```

Specifying an Owner

Instead of using the default owner name for the Sybase tables as defined in Sybase, or explicitly specifying a different owner in the SQL statements, you can specify a default owner that is used whenever a name is not explicitly specified in the SQL statements.

To specify the owner, set the gateway initialization parameter `HS_FDS_DEFAULT_OWNER` in the initialization parameter file.

See Also: [Appendix C, "Heterogeneous Services Initialization Parameters"](#) for information about the `HS_FDS_DEFAULT_OWNER` initialization parameter.

Creating Database Links

Any Oracle client connected to the Oracle database server can access Sybase data through the gateway. The Oracle client and the Oracle database server can reside on different machines. The gateway accepts connections only from the Oracle database server.

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

same database link and get a distinct connection to the gateway and Sybase database.

Database links are active for the duration of a gateway session. If you want to close a database link during a session, you can do so with the ALTER SESSION statement. The database and application administrators of a distributed database system are responsible for managing the necessary database links that define paths to the Sybase database.

See Also: *Oracle8i Administrator's Guide* and *Oracle8i Distributed Database Systems* for more information about using database links.

Configuring the Gateway for Multiple Sybase Databases

Note: This section describes how to configure the gateway to access two Sybase databases created in the same Sybase server. The examples given here can be used to access different Sybase servers, provided that the SYBASE environment variable in each initialization parameter file is the same. If you must have different values for the SYBASE environment variable in each initialization parameter file, then you must start different listener processes. Refer to *Net8 Administrator's Guide* for more information about starting multiple listener processes for a single Oracle instance.

The tasks for configuring the gateway to access multiple Sybase databases are similar to the tasks for configuring the gateway for a single database. The configuration example assumes the following:

- The gateway is installed and configured with the default SID of tg4sybs
- The ORACLE_HOME environment variable is set to the directory where the gateway is installed
- The gateway is configured for one Sybase database named db1
- Two Sybase databases named db2 and db3 on a server named syb115_sun are being added

Configuring the gateway for additional Sybase databases is similar to configuring it for one database, and involves the following:

- Configuring the gateway
- Configuring the Sybase environment

- Configuring Net8 for the gateway and the Oracle database server

Multiple Databases Example: Configuring the Gateway

Choose Two System IDs for Each Sybase Database

A separate instance of the gateway accesses the different Sybase databases. Each instance needs its own gateway System ID (SID). For this example, the gateway SIDs are chosen for the instances that access the Sybase databases:

- tg4sybs2 for the gateway accessing database db2
- tg4sybs3 for the gateway accessing database db3

Create Two Initialization Parameter Files

Create an initialization parameter file for each instance of the gateway by copying the original initialization parameter file,

`$ORACLE_HOME/tg4sybs/admin/inittg4sybs.ora`, twice, naming one with the gateway SID for db2 and the other with the gateway SID for db3:

```
$ cd $ORACLE_HOME/tg4sybs/admin
$ cp inittg4sybs.ora inittg4sybs2.ora
$ cp inittg4sybs.ora inittg4sybs3.ora
```

Change the value of the `HS_FDS_CONNECT_INFO` parameter in the new files.

For `inittg4sybs2.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=syb115_sun.db2
```

For `inittg4sybs3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=syb115_sun.db3
```

Note: If you have multiple gateway SIDs for the same Sybase database because you want to use different gateway parameter settings at different times, follow the same procedure. You create several initialization parameter files, each with different SIDs and different parameter settings.

Multiple Databases Example: Configuring the Sybase Environment

Set the Sybase environment variables in both of the new initialization parameter files, as follows:

For `inittg4sybs2.ora`, enter the following:

```
SET SYBASE=sybase_dir
```

For `inittg4sybs3.ora`, enter the following:

```
SET SYBASE=sybase_dir
```

sybase_dir is the directory that was specified for the `LD_LIBRARY_PATH`.

Multiple Databases Example: Configuring Oracle Net8 Listener

Add Entries to `listener.ora`

Add two new entries to the TNS listener configuration file, `listener.ora`. You must have an entry for each gateway instance, even when multiple gateway instances access the same database.

The following example shows the entry for the original installed gateway first, followed by the new entries:

```
SID_LIST_LISTENER=
(SID_LIST=
  (SID_DESC=
    (SID_NAME=tg4sybs)
    (ORACLE_HOME=oracle_home_directory)
    (PROGRAM=tg4sybs)
  )
  (SID_DESC=
    (SID_NAME=tg4sybs2)
    (ORACLE_HOME=oracle_home_directory)
    (PROGRAM=tg4sybs)
  )
  (SID_DESC=
    (SID_NAME=tg4sybs3)
    (ORACLE_HOME=oracle_home_directory)
    (PROGRAM=tg4sybs)
  )
)
```

Multiple Databases Example: Stopping and Starting the TNS Listener

The LD_LIBRARY_PATH environment variable must be set to include the Sybase libraries and the TNS listener must be started to initiate the new settings.

If you have the Bourne or Korn Shell, to set the LD_LIBRARY_PATH enter the following:

```
$ set LD_LIBRARY_PATH=$SYBASE/lib:$ORACLE_HOME/lib:$LD_LIBRARY_PATH;export LD_LIBRARY_PATH
```

If you have the C Shell, enter the following:

```
$ setenv LD_LIBRARY_PATH $SYBASE/lib:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Note: SYBASE is a Sybase environment variable. See your Sybase documentation for information about setting the environment variable to access the Sybase database.

If the listener is already running, use the lsnrctl command to stop the listener and then start it with the new settings, as follows:

```
$ lsnrctl stop
$ lsnrctl start
```

Multiple Databases Example: Configuring the Oracle Database Server for Gateway Access

Configuring Net8 on the Oracle Database Server for Multiple Gateway Instances

Add two connect descriptor entries to the `tnsnames.ora` file. You must have an entry for each gateway instance, even if the gateway instances access the same database.

The following Sybase example shows the entry for the original installed gateway first, followed by the two entries for the new gateway instances:

```
old_db_link=(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=TCP)
    (PORT=1541)
    (HOST=gtwhost))
  (CONNECT_DATA=
    (SID=tg4sybs))
```

```
                (HS=OK))
new_db2_link=(DESCRIPTION=
                (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                (SID=tg4sybs2))
                (HS=OK))
new_db3_link=(DESCRIPTION=
                (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                (SID=tg4sybs3))
                (HS=OK))
```

The value for PORT is the TCP/IP port number of the TNS listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the TNS listener. The value for HOST is the name of the machine on which the gateway is running. The name also can be found in the `listener.ora` file used by the TNS listener.

Multiple Databases Example: Accessing Sybase Data

Enter the following to create a database link for the `tg4sybs2` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK SYBS2 CONNECT TO
      2 user2 IDENTIFIED BY password2 USING 'tg4sybs2';
```

Enter the following to create a database link for the `tg4sybs3` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK SYBS3 CONNECT TO
      2 user3 IDENTIFIED BY password3 USING 'tg4sybs3';
```

After the database links are established you can query the new Sybase databases, as in the following:

```
SQL> SELECT * FROM ALL_USERS@SYBS2;
```

or

```
SQL> SELECT * FROM ALL_USERS@SYBS3;
```

Using the Gateway

After the gateway is installed and configured, you can use the gateway to access Sybase data, pass Sybase commands from applications to the Sybase database, perform distributed queries, and copy data.

This chapter contains the following sections:

- [Accessing Sybase Data Through the Gateway](#)
- [Oracle Database Server SQL Construct Processing](#)
- [Data Type Conversion](#)
- [Using Synonyms](#)
- [Using the Pass-Through Feature](#)
- [Executing Stored Procedures and Functions](#)
- [Performing Distributed Queries](#)
- [Copying Data from the Oracle Database Server to Sybase](#)
- [Copying Data from Sybase to the Oracle Database Server](#)
- [Monitoring Gateway Sessions](#)

Accessing Sybase Data Through the Gateway

To access Sybase data through the gateway, complete the following steps on the Oracle database server:

1. Log in to the Oracle database server.
2. Create a database link to the Sybase database, as in the following:

```
SQL> CREATE DATABASE LINK SYBS
      2   CONNECT TO user_id IDENTIFIED BY password
      3   USING 'SYBSDATA'
```

where:

user_id is the user ID used to establish a session in the Sybase. This user ID must be a valid Sybase database user ID and be authorized to use any database object in the Sybase database that is referenced in the SQL commands.

If *user_id* contains lowercase letters or non-alphanumeric characters, surround *user_id* with double quotation marks.

password is the password used to establish a session in the Sybase database. This must be a valid Sybase database password.

If *password* contains lowercase letters or non-alphanumeric characters, surround *password* with double quotation marks.

See Also: ["Database Links"](#) on page 1-13 and ["Creating Database Links"](#) on page 4-11 for more information.

3. Retrieve data from the Sybase database using one of the following methods:
 - If the CONNECT TO clause of the database link specifies ORACLE as the user ID, the following example retrieves data from the EMP table in the Sybase database using the name ORACLE as the Sybase database user:

```
SQL> SELECT * FROM "EMP"@SYBS
```

- The following SELECT statement retrieves the EMP table in the CORPDATA schema, using the name ORACLE as the Sybase database user:

```
SQL> SELECT * FROM "CORPDATA"."EMP"@SYBS
```

The ORACLE user must have the appropriate Sybase privileges to access the CORPDATA.EMP table.

4. Update data in the Sybase database using the UPDATE statement, as in the following example:

```
SQL> UPDATE "EMP"@SYBS SET "SAL"="SAL"*1.10
```

Oracle Database Server SQL Construct Processing

The gateway rewrites SQL statements when the statements need to be translated or post-processed.

For example, consider a program that requests the following from the Sybase database:

```
SELECT "COL_A" FROM "test"@SYBS  
WHERE "COL_A" = INITCAP('jones');
```

The Sybase database does not recognize INITCAP, so the Oracle database server does a table scan of test and filters the results locally. The gateway rewrites the SELECT statement as follows:

```
SELECT "COL_A" FROM "test"@SYBS
```

The results of the query are sent to the gateway and are filtered by the Oracle database server.

Consider the following UPDATE request:

```
UPDATE "test"@SYBS WHERE "COL_A" = INITCAP('jones');
```

In this case, the Oracle database server and the gateway cannot compensate for the lack of support at the Sybase side, so an error is issued.

If you are performing operations on large amounts of data stored in the Sybase database, keep in mind that some functions require data to be moved to the integrating Oracle database server before processing can occur.

Data Type Conversion

The gateway converts Sybase data types to Oracle data types as follows:

Table 5–1 Data Type Conversions

Sybase	Oracle	Comment
BINARY	RAW	
BIT	NUMBER(3)	
CHAR	CHAR	
DATETIME	DATE	Sybase DATETIME does not support the fractional part of the data
DECIMAL	NUMBER(18)	
DOUBLE PRECISION	FLOAT(49)	
FLOAT	FLOAT(49)	
IMAGE	LONG RAW	
INT	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
MONEY	NUMBER(p[,s])	
NCHAR	Not supported	
NUMERIC	NUMBER(p[,s])	
NVARCHAR	Not supported	
REAL	FLOAT(23)	
SMALLDATETIME	DATE	Sybase DATETIME does not support the fractional part of the data
SMALLINT	NUMBER(5)	NUMBER range is -32,767 to 32,767
SMALL MONEY	NUMBER(p[,s])	
TEXT	LONG	
TINYINT	NUMBER(3)	
VARBINARY	RAW	
VARCHAR	VARCHAR2	

Using Synonyms

You can provide complete data location transparency and network transparency by using the synonym feature of the Oracle database server. When a synonym is defined, you do not have to know the underlying table or network protocol. A synonym can be public, which means that all Oracle users can refer to the synonym. A synonym can also be defined as private, which means every Oracle user must have a synonym defined to access the Sybase table.

The following statement creates a system wide synonym for the EMP table in the schema of user ORACLE in the Sybase database:

```
SQL> CREATE PUBLIC SYNONYM EMP FOR "ORACLE"."EMP"@SYBS
```

See Also: *Oracle8i Administrator's Guide* for information about synonyms.

Using the Pass-Through Feature

The gateway can pass Sybase commands or statements from the application to the Sybase database using the DBMS_HS_PASSTHROUGH package.

Use the DBMS_HS_PASSTHROUGH package in a PL/SQL block to specify the statement to be passed to the Sybase database, as follows:

```
DECLARE
    num_rows INTEGER;
BEGIN
    num_rows := DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@SYBS( 'command' );
END;
/
```

Where *command* cannot be one of the following:

- BEGIN TRANSACTION
- COMMIT
- ROLLBACK
- SAVE
- SHUTDOWN
- Sybase tool commands

The DBMS_HS_PASSTHROUGH package does support passing bind values and executing SELECT statements.

See Also: *Oracle8i Distributed database Systems* for more information about the DBMS_HS_PASSTHROUGH package and the different functions supported by this package.

Executing Stored Procedures and Functions

Using the procedural feature, the gateway can execute stored procedures that are defined in the Sybase database. It is not necessary to relink the gateway or define the procedure to the gateway, but the procedure's access privileges must permit access by the gateway.

See Also: *Oracle8i Distributed Database Systems* for more information about executing stored procedures.

Standard PL/SQL statements are used to execute a stored procedure. By default, all stored procedures and functions do not return a return value to the user. To enable return values, set the HS_FDS_PROC_IS_FUNC parameter in the initialization parameter file.

See Also: For information about editing the initialization parameter file, see ["Task 2: Customize the Initialization Parameter File"](#) on page 4-2. For information about the HS_FDS_PROC_IS_FUNC parameter, see [Appendix C, "Heterogeneous Services Initialization Parameters"](#).

Note: If you set the HS_FDS_PROC_IS_FUNC gateway initialization parameter, you must change the syntax of the procedure execute statement for all existing stored procedures.

In the following example, the employee name JOHN SMYTHE is passed to the Sybase stored procedure REVISE_SALARY. The stored procedure retrieves the salary value from the Sybase 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';
    REVISE_SALARY@SYBS(INPUT,RESULT);
    UPDATE EMP SET SAL = RESULT WHERE ENAME =: INPUT;
END;
/

```

The procedural feature automatically converts non-Oracle data types to and from PL/SQL data types.

Stored procedures that return a result set are executed. However, any result sets are skipped and only output parameters and return values are returned.

Performing Distributed Queries

Oracle Transparent Gateway technology enables the execution of distributed queries that join the Oracle database server and the Sybase database, and any other data store for which Oracle Corporation provides a gateway. These complex operations can be invisible to the users requesting the data.

Example of a Distributed Query

The following example joins data between the Oracle database server, an IBM DB2 database, and the Sybase database:

```

SQL> SELECT O.CUSTNAME, P.PROJNO, E.ENAME, SUM(E.RATE*P."HOURS")
        FROM ORDERS@DB2 O, EMP@ORACLE8 E, "PROJECTS"@SYBS P
        WHERE O.PROJNO = P."PROJNO"
        AND P."EMPNO" = E.EMPNO
        GROUP BY O.CUSTNAME, P."PROJNO", E.ENAME

```

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

```

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

```

```
GROUP BY O.CUSTNAME, P."PROJNO", E.ENAME
```

Use the following SQL statement to retrieve information from the data stores in one command:

```
SQL> SELECT * FROM DETAILS;
```

The command retrieves the following table:

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

Two-Phase Commit

When the gateway is configured as COMMIT_CONFIRM, it is always the commit point site when the Sybase database is updated by the transaction, regardless of the HS_COMMIT_POINT_STRENGTH setting. The Oracle database server commits the unit of work in the Sybase database after verifying that all Oracle databases in the transaction have successfully prepared the transaction.

See Also: *Oracle8i Administrator's Guide* and *Oracle8i Distributed Database Systems* for more information about the two-phase commit process.

Only one gateway can participate in an Oracle two-phase commit transaction. Two-phase commit transactions are recorded in the HS_TRANSACTION_LOG table that is created when configuring the gateway.

See Also: ["Configuring for Two-Phase Commit"](#) on page 4-9.

Distributed Sybase Transactions

Because the HS_TRANSACTION_LOG table is used to record the status of a gateway transaction, the table must reside in the database where the Sybase update takes place.

Updates to the HS_TRANSACTION_LOG table cannot be part of an Oracle distributed transaction.

See Also: [Chapter 4, "Configuring the Gateway"](#) for more information.

Copying Data from the Oracle Database Server to Sybase

Use the SQL*Plus COPY command to copy data from the local database to the Sybase database. The syntax is as follows:

```
COPY FROM username/password@db_name
INSERT destination_table USING query
```

The following example selects all rows from the local Oracle EMP table, inserts them into the EMP table on the Sybase database, and commits the transaction:

```
SQL> COPY FROM SCOTT/TIGER@ORACLE8 -
> INSERT SCOTT.EMP@SYBS -
> USING SELECT * FROM EMP
```

The COPY command supports APPEND, CREATE, INSERT, and REPLACE options. However, INSERT is the only option supported when copying to Sybase. The SQL*Plus COPY command does not support copying to tables with lowercase table names. Use the following PL/SQL syntax with lowercase table names:

```
DECLARE
    v1 oracle_table.column1%TYPE;
    v2 oracle_table.column2%TYPE;
    v3 oracle_table.column3%TYPE;
    .
    .
    .
    CURSOR cursor_name IS SELECT * FROM oracle_table;
BEGIN
    OPEN cursor_name;
    LOOP
        FETCH cursor_name INTO v1, v2, v3, ... ;
        EXIT WHEN cursor_name%NOTFOUND;
        INSERT INTO destination_table VALUES (v1, v2, v3, ...);
    END LOOP;

    CLOSE cursor_name;
END;
```

```
/
```

See Also: *SQL*Plus User's Guide and Reference* for more information about the COPY command.

The following Oracle SQL INSERT statement is not supported for copying data from the Oracle database server to Sybase:

```
INSERT INTO table_name SELECT column_list FROM table_name
```

For example, consider the following statement:

```
SQL> INSERT INTO SYBS_TABLE SELECT * FROM MY_LOCAL_TABLE
```

The statement returns the following error message:

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

Copying Data from Sybase to the Oracle Database Server

The CREATE TABLE command lets you copy data from a Sybase database to the Oracle database server. To create a table on the local database and insert rows from the Sybase table, use the following syntax:

```
CREATE TABLE table_name AS query
```

The following example creates the table EMP in the local Oracle database and inserts the rows from the EMP table of the Sybase database:

```
SQL> CREATE TABLE EMP AS SELECT * FROM SCOTT."EMP"@SYBS
```

Alternatively, you can use the SQL*Plus COPY command to copy data from the Sybase database to the Oracle database server.

See Also: *SQL*Plus User's Guide and Reference* for more information about the COPY command.

Monitoring Gateway Sessions

The Distributed Access Manager is used for monitoring distributed sessions that involve non-Oracle systems. The tool is distributed and installed with Oracle Enterprise Manager and runs as a standalone application.

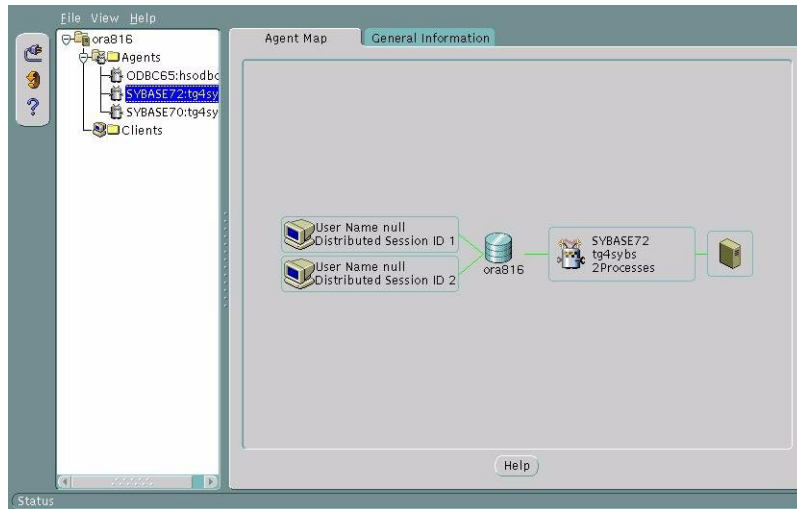
Start the Distributed Access Manager as follows:

```
oemapp damgr
```


Using Distributed Access Manager requires connecting to an instance of an Oracle database. Once connected, the tool displays all of the gateways currently registered with the server. For each active gateway, runtime information about the gateway is displayed. It contains the machine on which the gateway runs, its process ID, and the operating system user who started the gateway. The modules supported in the gateway are also listed.

In addition, data about the current sessions through this gateway are available. Sessions are tied to the gateway by the gateway's class and instance ID, which uniquely identify the gateway. The parent client session of the distributed session is described in terms of the client program, the user/schema that owns the session, the session ID, the status, and the terminal.

Figure 5–1 Distributed Access Manager View of a Session



Information about the database link that was used for establishing the distributed connection, the startup time, and the set of initialization parameters used for the session is also available.

All of the runtime information is derived from dynamically updated V\$ tables. The Distributed Access Manager has a refresh capability available through the menu and toolbar that allows users to rerun queries if necessary and update the data.

When the data is refreshed, the tool verifies that the set of registered agents remains the same. If it is not, the global view is updated.

See Also: *Oracle Enterprise Manager Administrator's Guide* and online help for more information about the Distributed Access Manager.

Case Studies

The following case studies for Sybase demonstrate some of the features of the Oracle Transparent Gateway. You can verify that the gateway is installed and operating correctly by using the demonstration files included on the distribution CD-ROM.

The demonstration files are automatically copied to disk when the gateway is installed.

This chapter contains the following sections:

- [Case Descriptions](#)
- [CD-ROM Contents](#)
- [Demonstration Files](#)
- [Demonstration Requirements](#)
- [Creating Demonstration Tables](#)
- [Case 1: Simple Queries](#)
- [Case 2: A More Complex Query](#)
- [Case 3: Joining Sybase Tables](#)
- [Case 4: Write Capabilities](#)
- [Case 5: Data Dictionary Query](#)
- [Case 6: The Pass-Through Feature](#)
- [Case 7: Executing Stored Procedures](#)

Case Descriptions

The cases illustrate:

- A simple query (Case 1)
- A more complex query (Case 2)
- Joining Sybase tables (Case 3)
- Write capabilities (Case 4)
- A data dictionary query (Case 5)
- The pass-through feature (Case 6)
- Executing a stored procedure (Case 7)

CD-ROM Contents

The distribution CD-ROM contains the following:

- Demonstration files
- One SQL script file that creates the demonstration tables and stored procedures in the Sybase database
- One SQL script file that drops the demonstration tables and stored procedures from the Sybase database

Demonstration Files

After a successful gateway installation, use the demonstration files stored in the directory `$ORACLE_HOME/tg4sybs/demo` where `$ORACLE_HOME` is the `$ORACLE_HOME` directory under which the gateway is installed. The directory contains the following demonstration files:

<code>bldsybs.sql</code>	<code>case5.sql</code>
<code>case1.sql</code>	<code>case6a.sql</code>
<code>case2.sql</code>	<code>case6b.sql</code>
<code>case3.sql</code>	<code>case7.sql</code>
<code>case4a.sql</code>	<code>dropsybs.sql</code>
<code>case4b.sql</code>	
<code>case4c.sql</code>	

Demonstration Requirements

The case studies assume these requirements have been met:

- The gateway demonstration tables and stored procedures are installed in the Sybase database
- The Oracle server has an account named SCOTT with a password of TIGER
- The Oracle server has a database link called GTWLINK (set up as public or private to the user SCOTT) which connects the gateway to a Sybase database as SCOTT with password TIGER2

For example, you can create the database link as follows:

```
SQL> CREATE DATABASE LINK GTWLINK CONNECT TO SCOTT
      2 IDENTIFIED BY TIGER2 USING 'GTWSID';
```

- Net8 is configured correctly and running
- The Sybase environment variable, SYBASE, is set correctly

Creating Demonstration Tables

The case studies are based on the GTW_EMP, GTW_DEPT, and GTW_SALGRADE tables and the stored procedures InsertDept and GetDept. If the demonstration tables and stored procedures have not been created in the Sybase database, use the bldsybs.sql script to create them. Enter the following:

```
$ isql -USCOTT -PTIGER2 -ibldsybs.sql
```

The script creates the demonstration tables and stored procedures in the Sybase database accordingly:

```
CREATE TABLE GTW_EMP (
EMPNO      SMALLINT NOT NULL
ENAME      VARCHAR(10),
JOB        VARCHAR(9),
MGR        SMALLINT,
HIREDATE   DATETIME,
SAL        NUMERIC(7,2),
COMM       NUMERIC(7,2),
DEPTNO     SMALLINT)
go
```

```
CREATE TABLE GTW_DEPT (  
  DEPTNO      SMALLINT NOT NULL,  
  DNAME       VARCHAR(14),  
  LOC        VARCHAR(13))  
go  
  
CREATE TABLE GTW_SALGRADE (  
  GRADE       MONEY,  
  LOSAL       NUMERIC(9,4),  
  HISAL       NUMERIC(9,4))  
go  
  
DROP PROCEDURE InsertDept  
go  
  
CREATE PROCEDURE InsertDept (@dno INTEGER,  
  @dname VARCHAR(14), @loc VARCHAR(13))  
AS INSERT INTO GTW_DEPT VALUES (@dno, @dname, @loc)  
go  
  
DROP PROCEDURE GetDept  
go  
  
CREATE PROCEDURE GetDept (@dno INTEGER, @dname VARCHAR(14) OUTPUT)  
AS SELECT @dname=DNAME FROM GTW_DEPT WHERE DEPTNO=@dno  
go
```

Demonstration Table Definitions

The table definitions are listed below using information retrieved by the SQL SERVER MANAGER DESCRIBE command:

GTW_EMP

Name	Null?	Type
-----	-----	----
EMPNO	NOT NULL	NUMBER(5)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(5)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(5)

GTW_DEPT

Name	Null?	Type
-----	-----	-----
DEPTNO	NOT NULL	NUMBER(5)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

GTW_SALGRADE

Name	Null?	Type
-----	-----	-----
GRADE		NUMBER(19,4)
LOSAL		NUMBER(9,4)
HISAL		NUMBER(9,4)

Demonstration Table Contents

The contents of the Sybase tables are:

GTW_EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
-----	-----	---	---	-----	---	----	-----
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

GTW_DEPT

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK

20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

GTW_SALGRADE

GRADE	LOSAL	HISAL
-----	-----	-----
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Case 1: Simple Queries

Case 1 demonstrates the following:

- A simple query
- A simple query retrieving full date information

The first query retrieves all the data from GTW_DEPT and confirms that the gateway is working correctly. The second query retrieves all the data from GTW_EMP including the time portion of the hire date because the default date format was set to DD-MON-YY HH24:MM:SS for the session by an ALTER SESSION command.

To run Case 1, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE1
```

Case 1 executes two SQL statements. The first statement is as follows:

```
SELECT * FROM GTW_DEPT@GTWLINK;
```

which results in the following:

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

The following command and SQL statement change the date format to DD-MON-YY HH24:MM:SS in Oracle and retrieve the employee name and hire date from the Sybase database:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YY HH24:MM:SS';
SELECT ENAME, HIREDATE FROM GTW_EMP@GTWLINK;
```

which results in the following:

ENAME	HIREDATE
SMITH	17-Dec-80 00:00:00
ALLEN	20-Feb-81 00:00:00
WARD	22-Feb-81 00:00:00
JONES	02-Apr-81 00:00:00
MARTIN	28-Sep-81 00:00:00
BLAKE	01-May-81 00:00:00
CLARK	09-Jun-81 00:00:00
SCOTT	09-Dec-82 00:00:00
KING	17-Nov-81 00:00:00
TURNER	08-Sep-81 00:00:00
ADAMS	12-Jan-83 00:00:00
JAMES	03-Dec-81 00:00:00
FORD	03-Dec-81 00:00:00
MILLER	23-Jan-82 00:00:00

14 rows selected.

Case 2: A More Complex Query

Case 2 demonstrates the following:

- The functions `SUM(expression)` and `NVL(expr1, expr2)` in the SELECT list
- The GROUP BY and HAVING clauses

This query retrieves the departments from GTW_EMP whose total monthly expenses are higher than \$10,000.

To run Case 2, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE2
```

which executes the following SQL statement:

```
SELECT SUM(SAL), SUM(COMM), SUM(SAL + NVL(COMM,0))
"TOTAL MONTHLY EXPENSES OF", DEPTNO "DEPARTMENT"
```

```
FROM GTW_EMP@GTWLINK GROUP BY DEPTNO
HAVING SUM(SAL + NVL(COMM,0)) > 10000;
```

which results in the following:

SUM(SAL)	SUM(COMM)	TOTAL MONTHLY EXPENSES OF DEPARTMENT	
10875		10875	20
9400	2200	11600	30

Case 3: Joining Sybase Tables

Case 3 demonstrates the following:

- Joins between Sybase tables
- Subselects
- SQL tracing

The query retrieves information from three Sybase tables and relates the employees to their department name and salary grade, but only for those employees earning more than the average salary.

To run Case 3, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE3
```

which executes the following SQL statement:

```
SELECT ENAME, DNAME, GRADE
FROM GTW_EMP@GTWLINK A, GTW_DEPT@GTWLINK B,
GTW_SALGRADE@GTWLINK C
WHERE A.DEPTNO = B.DEPTNO
AND SAL >= LOSAL AND SAL <= HISAL
AND SAL > (SELECT AVG(SAL) FROM GTW_EMP@GTWLINK);
```

which results in the following:

ENAME	DNAME	GRADE
CLARK	ACCOUNTING	4
KING	ACCOUNTING	5
JONES	RESEARCH	4
SCOTT	RESEARCH	4
FORD	RESEARCH	4

```
BLAKE      SALES      4
6 rows selected.
```

Case 4: Write Capabilities

Case 4 is split into three cases and demonstrates the following:

- [DELETE Statement](#)
- [UPDATE Statement](#)
- [INSERT Statement](#)

DELETE Statement

Case 4a demonstrates bind values and subselect. All employees in department 20 and one employee, WARD, in department 30 are deleted.

To run Case 4a, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE4A
```

which executes the following PL/SQL block:

```
DECLARE
    EID          SMALLINT;
    EMPLOYEE     VARCHAR(10);
    DEPARTMENT   VARCHAR(14);
BEGIN
    EMPLOYEE := 'WARD';
    DEPARTMENT := 'RESEARCH';
    SELECT EMPNO INTO EID FROM GTW_EMP@GTWLINK
    WHERE ENAME = EMPLOYEE;
    DELETE FROM GTW_EMP@GTWLINK
    WHERE EMPNO = EID
    OR DEPTNO = (SELECT DEPTNO FROM GTW_DEPT@GTWLINK
    WHERE DNAME = DEPARTMENT);
END;
/
```

Note: For Sybase, you must add the schema name before the dblink. For example, instead of GTW_DEPT@GTWLINK, you need SCOTT.GTW_DEPT@GTWLINK.

To ensure the outcome is correct, the script also issues:

```
SELECT ENAME, DEPTNO FROM GIW_EMP@GIWLINK;  
ROLLBACK;
```

which results in the following:

ENAME	DEPTNO
-----	-----
ALLEN	30
MARTIN	30
BLAKE	30
CLARK	10
KING	10
TURNER	30
JAMES	30
MILLER	10

8 rows selected.

Rollback complete.

WARD and all employees of the research department 20 were removed.

UPDATE Statement

Case 4b provides an example of a simple UPDATE statement. In this example, employees are given a \$100 per month salary increase. To run Case 4b, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE4B
```

This script issues these statements:

```
UPDATE GIW_EMP@GIWLINK SET SAL = SAL + 100;  
SELECT ENAME, SAL FROM GIW_EMP@GIWLINK;  
ROLLBACK;
```

which results in the following:

ENAME	SAL
-----	-----
SMITH	900
ALLEN	1700
WARD	1350
JONES	3075

MARTIN	1350
BLAKE	2950
CLARK	2550
SCOTT	3100
KING	5100
TURNER	1600
ADAMS	1200
JAMES	1050
FORD	3100
MILLER	1400

14 rows selected.

Rollback complete.

INSERT Statement

Case 4c is an example of a simple insert statement that does not provide information for all columns. To run Case 4c, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE4C
```

The script adds the department SHIPPING as department 50 to GTW_DEPT by issuing the following:

```
INSERT INTO GTW_DEPT@GTWLINK (DEPTNO, DNAME)
VALUES (50, 'SHIPPING');
SELECT * FROM GTW_DEPT@GTWLINK;
ROLLBACK;
```

which results in the following:

1 row created.

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	SHIPPING	

Rollback complete.

Case 5: Data Dictionary Query

Case 5 demonstrates data dictionary mapping. It retrieves all the tables and views that exist in the Sybase database that begin with "GTW". To run Case 5, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE5
```

The script issues the following statement:

```
SELECT * FROM ALL_TABLES@GTWLINK
WHERE SUBSTR(TABLE_NAME,1,3) = 'GTW';
```

which results in the following:

OWNER	TABLE_NAME										T	C	I	PCT_FREE					

PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS														

MAX_EXTENTS	PCT_INCREASE	FREELISTS	FREELIST_GROUPS	L	B	NUM_ROWS	BLOCKS												

EMPTY_BLOCKS	AVG_SPACE	CHAIN_CNT	AVG_ROW_LEN	AVG_SPACE_FREELIST_BLOCKS															

NUM_FREELIST_BLOCKS	D	I	C	T	SAMPLE_SIZE	L	P	I	T	S	N	B	R	G	U	D	S	M	

SCOTT	GTW_EMP																		0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SCOTT	GTW_DEPT																		0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SCOTT	GTW_SALGRADE																		0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SCOTT	GTW_BONUS																		0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

```

      0      0      0      0      0
        0          0
SCOTT      GTW_DUMMY      0
      0      0      0      0      0      0      0
      0      0      0      0      0      0      0
      0      0      0      0      0

```

Case 6: The Pass-Through Feature

Case 6 demonstrates the gateway pass-through feature which allows an application to send commands or statements to Sybase.

This case demonstrates:

- A pass-through UPDATE statement using bind variables
- A pass-through SELECT statement

UPDATE Statement

Case 6a provides an example of a pass-through UPDATE statement with bind variables. In this example, the salary for EMPNO 7934 is set to 4000. To run Case 6a, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE6A
```

The script issues these statements:

```

SELECT * FROM GTW_EMP@GTWLINK WHERE EMPNO = 7934;

DECLARE
  crs binary_integer;
  ret integer;
BEGIN
  crs := DBMS_HS_PASSTHROUGH.OPEN_CURSOR@GTWLINK;
  DBMS_HS_PASSTHROUGH.PARSE@GTWLINK(crs,
    'UPDATE GTW_EMP SET SAL = ? WHERE EMPNO = ?');
  DBMS_HS_PASSTHROUGH.BIND_VARIABLE@GTWLINK(crs, 1, 4000);
  DBMS_HS_PASSTHROUGH.BIND_VARIABLE@GTWLINK(crs, 2, 7934);
  ret := DBMS_HS_PASSTHROUGH.EXECUTE_NON_QUERY@GTWLINK(crs);
  DBMS_HS_PASSTHROUGH.CLOSE_CURSOR@GTWLINK(crs);
END;
/

```

```
SELECT * FROM GTW_EMP@GTWLINK WHERE EMPNO = 7934;
```

The results are as follows:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

DEPTNO						

7934	MILLER	CLERK	7782	23-JAN-82	1300	
10						

PL/SQL procedure successfully completed.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

DEPTNO						

7934	MILLER	CLERK	7782	23-JAN-82	4000	
10						

SELECT Statement

Case 6b provides an example of a pass-through SELECT statement. The data that is returned from the SELECT statement is inserted into a local table at the Oracle database server. To run Case 6b, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE6B
```

The first two statements drop and recreate the LOCAL_PT_TABLE table in the Oracle database. The PL/SQL block uses the DBMS_HS_PASSTHROUGH package to execute a SELECT statement at Sybase and inserts the data returned into LOCAL_PT_TABLE:

```
DROP TABLE LOCAL_PT_TABLE;
CREATE TABLE LOCAL_PT_TABLE (EMPNO NUMBER(5), ENAME VARCHAR2(10));

DECLARE
  crs binary_integer;
  ret binary_integer;
  ename varchar2(10);
  empno number;
```



```

BEGIN
  crs := DBMS_HS_PASSTHROUGH.OPEN_CURSOR@GTWLINK;
  DBMS_HS_PASSTHROUGH.PARSE@GTWLINK(crs,
    'SELECT EMPNO, ENAME FROM GTW_EMP');
  begin
    ret := 0;
    WHILE (true)
    LOOP
      ret := DBMS_HS_PASSTHROUGH.FETCH_ROW@GTWLINK(crs, false);
      DBMS_HS_PASSTHROUGH.GET_VALUE@GTWLINK(crs, 1, empno);
      DBMS_HS_PASSTHROUGH.GET_VALUE@GTWLINK(crs, 2, ename);
      insert into LOCAL_PT_TABLE values (empno, ename);
    END LOOP;
  EXCEPTION
    when NO_DATA_FOUND then
      BEGIN
        DBMS_OUTPUT.PUT_LINE('End of Fetch');
        DBMS_HS_PASSTHROUGH.CLOSE_CURSOR@GTWLINK(crs);
      END;
  END;
END;
/

SELECT * FROM LOCAL_PT_TABLE;

```

The results are as follows:

Table created.

PL/SQL procedure successfully completed.

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER

```
7876 ADAMS
7900 JAMES
7902 FORD
7934 MILLER
```

14 rows selected.

Case 7: Executing Stored Procedures

Case 7 demonstrates the gateway executing a stored procedure in the Sybase database.

To run Case7, log on to SQL*Plus as SCOTT/TIGER and enter the following:

```
SQL> START CASE7
```

The first two statements drop and recreate the LOCAL_GTW_DEPT table in the Oracle database:

```
DROP TABLE LOCAL_GTW_DEPT;
CREATE TABLE LOCAL_GTW_DEPT (DEPTNO INTEGER, DEPTNAME VARCHAR2(14));
```

The next PL/SQL block declares the variable DNAME and executes the stored procedure GetDept in the Sybase. The stored procedure retrieves the department name from the Sybase table, as follows:

```
DECLARE
    DNAME VARCHAR2(14);
BEGIN
    "GetDept"@GTWLINK(10, DNAME);
    INSERT INTO LOCAL_GTW_DEPT VALUES (10, DNAME);
END;
/
```

The results in this row in the Oracle LOCAL_GTW_DEPT table are as follows:

```
DEPTNO DNAME
-----
10 ACCOUNTING
```

Diagnosing Problems

This chapter provides information about diagnosing problems and understanding error messages. The following sections are included:

- [Problem Identification](#)
- [Installation and Configuration Issues](#)
- [Net8 Configuration Problems](#)
- [Common Gateway Problems](#)
- [Analyzing Error Messages](#)
- [Tracing](#)
- [Oracle Support Services](#)

Problem Identification

When you encounter a problem, you first should determine where the problem occurred. The following is a list of possible problem areas and references to more information about those areas.

- Installation and configuration

See Also: ["Installation and Configuration Issues"](#) on page 7-3.

- Net8

You might be experiencing problems in Net8 if you receive errors that appear as follows:

```
ORA-nnnnn: TNS: error_text
```

See Also: ["Net8 Configuration Problems"](#) on page 7-3.

- Oracle database server

You might experience problems with the Oracle database server if the database link is not set up correctly or if the Oracle database server does not have the distributed option. Verify the installation of the Oracle database server, and check the case (upper or lower) of the letters in the user name and password used in the CREATE DATABASE LINK statement.

See Also: ["Tracing"](#) on page 7-9.

- Sybase or the gateway

You might be experiencing problems with Sybase or with the gateway if you receive errors that appear as:

```
ORA-nnnnn: error_text  
[Transparent gateway for SYBASE]gateway_error_text
```

where *nnnnn* ranges from 12300 to 12399 or *error_text* involves the database link that calls the gateway.

See Also: ["Common Gateway Problems"](#) on page 7-5, ["Analyzing Error Messages"](#) on page 7-6 and ["Tracing"](#) on page 7-9.

To see if the error originated with Sybase, try the SQL statement with Sybase native tools. Verify that the SQL executed correctly and has no syntax errors. Contact the Sybase vendor for support.

Installation and Configuration Issues

The issues surrounding installation and configuration usually involve verifying if the installation process was successful, and wanting to know what to configure.

How do I verify that the installation succeeded?

Review the installation log file, which lists any problems encountered during installation. By default, the file has the following name:

```
orainventory_location/logs/installActions.log
```

where `orainventory_location` is the directory where the OraInventory is located.

Does the Oracle Installer configure everything?

No. The Oracle Installer simply installs the software. You must configure the software after its installation. Follow the tasks in [Chapter 4, "Configuring the Gateway"](#), to configure the gateway and its components.

Net8 Configuration Problems

If you receive errors that refer to TNS when accessing the gateway, the problem is probably with Net8 or the Net8 configuration. Following is a list of the most likely TNS errors and the actions you should take.

ORA-12154 "TNS:could not resolve service name"

Cause: The service name specified is not defined correctly in the `tnsnames.ora` file.

Action: Check the following items:

- Verify the `tnsnames.ora` file exists, is in the proper place, and is accessible.
- Verify the service name exists in one of the `tnsnames.ora` files. Add a service name if needed.

- Ensure there are no syntax errors in the `tnsnames.ora` file. Look for unmatched parentheses or stray characters. Any error in the `tnsnames.ora` file makes it unusable. If possible, regenerate the configuration files using the Oracle Network Manager.

See Also: *Net8 Administrator's Guide*.

- Ensure the USING clause of the database link you created for the Sybase database is pointing to the correct `tnsnames.ora` file.

See Also: ["Configuring the Gateway for Multiple Sybase Databases"](#) on page 4-12.

- If the `sqlnet.ora` file exists in the `$ORACLE_HOME/network/admin` directory, check to see if the file includes entries such as:

```
names.directory_path=(TNSNAMES, HOSTNAME)
names.default_domain=world
names.default_zone=world
```

If so, the service name might need the suffix `.world`.

ORA-12203 "TNS:unable to connect to destination"

Cause: An invalid TNS address was supplied or the destination is not listening. This error can also occur because of underlying network transport problems.

Action: For the Oracle database server:

- Ensure that the `tnsnames.ora` file is configured as described in ["Configuring the Gateway for Multiple Sybase Databases"](#) on page 4-12.
- Ensure that the USING clause of the database link you created is pointing to the correct `tnsnames.ora` file.

For the gateway:

- Ensure that the `listener.ora` on the gateway machine is configured as described in ["Configuring Net8 Listener for the Gateway"](#) on page 4-3.
- Ensure that the Net8 listener is started.

If the Net8 configuration is correct but you still encounter errors, set up a loop-back link, which is a database link in the Oracle database server that

points back to itself. If Net8 is working properly you can access a table in the Oracle database server using a loop-back link. For example:

```
SELECT * FROM table@loop
```

If this link does not enable you to access a table in the Oracle database server, see the Net8 documentation for the platform for information about troubleshooting Net8.

See Also: For more information about configuring the `listener.ora` and `tnsnames.ora` files, see ["Configuring Net8 Listener for the Gateway"](#) on page 4-3 and ["Configuring the Gateway for Multiple Sybase Databases"](#) on page 4-12.

ORA-12500 "TNS:listener failed to start a dedicated server process"

Cause: The process of starting the Oracle database server failed. The Oracle database server executable could not be found or the environment is set up incorrectly.

Action: Check these items:

- Verify that the SID value specified in the `tnsnames.ora` file matches the gateway SID.

Common Gateway Problems

Many common gateway problems are caused by errors in the initialization parameter file, and are described in this section. For other errors, review the log files for more information. If no log file exists, the gateway might not have started. In that case, Net8 configuration might be the problem.

See Also: ["Oracle Support Services"](#) on page 7-10.

ORA-02068: following severe error from *database_link_name*

ORA-01012: not logged error_found_in_init_file

Cause: One or more gateway initialization parameters might be configured incorrectly or there are syntax errors.

Action: Review the gateway log file for error messages that describe the problem. Correct the `inittg4sybs.ora` file as needed.

ORA-02068: following severe error from *database_link_name*

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

Cause: The HS_FDS_TRACE_LEVEL parameter is set to an invalid value in the initialization parameter file, or the target database is not set up correctly.

Action: Ensure the values are set correctly and the target database is set up correctly. Refer to the documentation for the target database.

ORA-02085: database link *database_link_name*. WORLD connects to *database_link_name.domain_name*

Cause: The Oracle database server has GLOBAL_NAMES set to TRUE in its initialization parameter file. This error occurs because the value specified by HS_DB_DOMAIN in the initialization parameter file, displayed as *domain_name* in the error message, does not match a domain name of WORLD.

Action: Change the value of HS_DB_DOMAIN in the initialization parameter file to WORLD.

ORA-02085: database link *database_link_name1.domain_name* connects to *database_link_name2.domain_name*

Cause: The Oracle database server has GLOBAL_NAMES set to TRUE in its initialization parameter file. The value specified by HS_DB_NAME in the initialization parameter file (*database_link_name1*) must match the value of the database link used by the gateway (*database_link_name2*). The error occurs because the values do not match.

Action: Change the value of HS_DB_NAME in the initialization parameter file to match the database link name being used.

Analyzing Error Messages

This section describes the following topics:

- [Message and Error Code Processing](#)
- [Oracle Database Server Error Codes](#)
- [Gateway Error Codes and Message Formats](#)

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 an SQL statement that refers to one or more Sybase 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.

Error conditions are represented in the following ways:

- An Oracle error code
- A gateway error code

Gateway error conditions are reported to the application using one of the gateway error codes in the range of ORA-28500 through ORA-28561.

See Also: ["Gateway Error Codes and Message Formats"](#) on page 7-8 for an explanation of the message format.

Oracle Database Server Error Codes

The Oracle database server messages and error codes are documented in *Oracle8i Server Messages*. Refer to that publication for information about interpreting the Oracle database server messages and for information about specific error codes.

If an error is detected by the Oracle database server, gateway message lines do not occur. For example, if the gateway cannot be accessed because of a Net8 or gateway installation problem, a gateway message line is not present in the generated error.

Another example of error messages without gateway message lines occurs when an INSERT statement attempts to insert data into a table, but does not include values for all of the columns in the table. The following SQL statement causes an error message:

```
SQL> insert into "EMP"@SYBS values(9999);
```

```
ERROR at line 1:  
ORA-00947: not enough values
```

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

Gateway Error Codes and Message Formats

Error codes are generally accompanied by additional message text, beyond the text associated with the Oracle message number. The additional text includes details about the error.

Gateway messages have the following format:

```
ORA-nnnnn: error_text  
[Transparent gateway for SYBASE]gateway_error_text
```

```
ORA-02063: preceding n lines from dblink
```

where:

<i>nnnnn</i>	is an Oracle error number where <i>nnnnn</i> ranges from 28500 to 28561.
<i>error_text</i>	is the text of the message associated with the error.
<i>gateway_error_text</i>	are additional messages generated by the gateway or by Sybase. See " Oracle Database Server Error Codes " on page 7-7 for more information.

The format of the *gateway_error_text* is:

	<pre>[Transparent gateway for SYBASE]<i>gateway_error_text</i></pre>
<i>n</i>	is the total number of gateway message lines.
<i>dblink</i>	is the name of the database link used to access the gateway.

For example, if you get a message where *nnnnn* is between 28500 and 28561, it is a gateway message and that is where the error is occurring. A gateway message line such as the following means that the error originates in Sybase or the gateway:

```
[Transparent gateway for SYBASE]
```

The ORA-28500 error code is returned for all errors for which a more specific error code does not exist. When the ORA-28500 error code is returned, the error might have been caused in the gateway by the Sybase components of the target database.

Tracing

Use error tracing to pinpoint problem areas involving the gateway, the Oracle database server, and Sybase.

You can enable error tracing to record by setting the HS_FDS_TRACE_LEVEL gateway initialization parameter.

See Also: For information, see [Appendix C, "Heterogeneous Services Initialization Parameters"](#).

Use tracing only while testing and debugging the application. Do not enable tracing when the application is running in a production environment because tracing reduces gateway performance.

Trace Log Files

A log file is created with each SQL session to collect the following information:

- The SQL statement as it was sent to the gateway and then on to the target database
- User information such as user name, user program and terminal ID

Trace files are created in the \$ORACLE_HOME/TG4SYBS/LOG directory, where \$ORACLE_HOME is the \$ORACLE_HOME directory under which the gateway is installed. The trace file names have the format *agent_sid_agt_process_id.trc* where *agent_sid* is the sid for the gateway and *process_id* is the operating system process ID of the gateway session.

Reviewing the Trace Output

If you are having trouble accessing data from the Sybase database, you might receive the following messages:

```
ORA-00942: table or view does not exist
[Transparent gateway for SYBASE]Object not found....
ORA-02063: preceding n lines from database_link_name
```

or the following message:

```
ORA-00904: invalid column name
```

To find the source of the problem, review the trace output from the gateway.

If you are certain that the identifier exists in the Sybase database, the trace output might show, for example, that the table or column name in the SQL statement sent to the Sybase database has the wrong case, or you forgot to enclose it in double quotation marks. The Oracle database server identifiers must be in uppercase unless you enclose them in double quotation marks. The case for identifiers for Sybase depends on whether it is case-sensitive or not.

The outgoing parsed SQL statement listed in the trace output is the SQL statement passed to the gateway and on to the target database. If you think the SQL statement might have been generated incorrectly, test the SQL statement by executing it with Sybase’s native tools or using an ODBC application such as ODBC-Test. If you need help to execute the statement using these tools, contact the Sybase vendor for support.

Oracle Support Services

If you determined that the problem does involve an Oracle component, but you have not been able to solve it, gather the information and fill out the worksheet before contacting Oracle Customer Support Services:

Note: Support for non-Oracle components is the responsibility of the vendor providing that component.

Table 7–1 Oracle Support Services Worksheet

Required Information	Site-specific Values
Name and full version numbers for:	
- the gateway	Oracle Transparent Gateway for Sybase Version: _____
- the platform/OS on which the gateway resides	Sun SPARC Solaris _____ Version: _____ _____ Machine Name: _____
Versions of Net8 Server used by the gateway	Net8 Version: _____
Net8 Adapter used by the gateway	Circle one: Named Pipes - SPX/IPX - TCP/IP Version: _____

Table 7-1 (Cont.) Oracle Support Services Worksheet

Required Information	Site-specific Values
Full pathname to directory in which gateway resides	Path: _____
Oracle database server and the platform/OS on which it resides	Oracle___ Server, Version: _____ OS: _____ Version: _____ Machine Name: _____
Oracle SID of the Oracle database server	SID: _____
Full pathname to directory where the Oracle database server is installed	ORACLE_HOME: _____
Version of Net8 Client used by the Oracle database server	Version: _____
Net8 adapter used by the Oracle database server	Circle one: Named Pipes - SPX/IPX - TCP/IP Version: _____
The database and any patch information for it	Server Name: _____ Database Name: _____ Version: _____
LD_LIBRARY_PATH	Path: _____
Listing of spool log if using SQL*Plus, or the SQL statement and error message received	Check here: <input type="checkbox"/>
Relevant table and view definition (a full description of the tables involved in the SQL query)	Check here: <input type="checkbox"/>
Description of the problem (input, output, test cases, and so on)	Check here: <input type="checkbox"/>
Listings of key files:	Check here: <input type="checkbox"/> tg4sybs_agt_pid.trc log file gathered with HS_FDS_TRACE_LEVEL <input type="checkbox"/> Net8 configuration files, tnsnames.ora and listener.ora <input type="checkbox"/> initialization parameter file

Supported SQL Syntax and Functions

This appendix contains the following sections:

- [Supported SQL Statements](#)
- [Oracle Functions](#)

Supported SQL Statements

With a few exceptions, the gateway provides full support for Oracle DELETE, INSERT, SELECT, and UPDATE statements.

The gateway does not support Oracle Data Definition Language (DDL) statements. No form of the Oracle ALTER, CREATE, DROP, GRANT, or TRUNCATE statements can be used. Instead, use the pass-through feature of the gateway if you need to use DDL statements against the Sybase database.

See Also: ["Using the Pass-Through Feature"](#) on page 5-5 and *Oracle8i SQL Reference* for a detailed descriptions of keywords, parameters, and options.

DELETE

The DELETE statement is fully supported. However, only Oracle functions supported by Sybase can be used.

See Also: ["Functions Supported by Sybase"](#) on page A-3 for a list of supported functions.

INSERT

The INSERT statement is fully supported. However, only Oracle functions supported by Sybase can be used.

See Also: ["Functions Supported by Sybase"](#) on page A-3 for a list of supported functions.

SELECT

The SELECT statement is fully supported, with these exceptions:

- CONNECT BY *condition*
- FOR UPDATE
- FOR UPDATE OF
- NOWAIT
- START WITH *condition*
- WHERE CURRENT OF

UPDATE

The UPDATE statement is fully supported. However, only Oracle functions supported by Sybase can be used. Also, you cannot have SQL statements in the subquery that refer to the same table name in the outer query. Subqueries are not supported in the SET clause

See Also: ["Functions Supported by Sybase"](#) on page A-3 for a list of supported functions.

Oracle Functions

All functions are evaluated by the Sybase database after the gateway has converted them to Sybase SQL.

Functions Not Supported by Sybase

Oracle SQL functions with no equivalent function in Sybase are not supported in DELETE, INSERT, or UPDATE statements, but are evaluated by the Oracle database server if the statement is a SELECT statement. That is, the Oracle database server performs post-processing of SELECT statements sent to the gateway.

If an unsupported function is used in a DELETE, INSERT, or UPDATE statement, the following Oracle error occurs:

ORA-02070: database *db_link_name* does not support *function* in this context

Functions Supported by Sybase

The gateway translates the following Oracle database server functions in SQL statements to their equivalent Sybase functions:

- [Arithmetic Operators](#)
- [Comparison Operators](#)
- [String Functions](#)
- [Group Functions](#)
- [Other Functions](#)

Arithmetic Operators

Oracle	Sybase
+	+
-	-
*	*
/	/

Comparison Operators

Oracle	Sybase
=	=
>	>
<	<
>=	>=
<=	<=
<>, !=, ^=	<>, !=
IS NOT NULL	IS NOT NULL
IS NULL	IS NULL

String Functions

Oracle	Sybase
	+ (expression1 + expression2)
ASCII	ASCII
CHR	CHAR
INSTR	CHARINDEX
LENGTH	CHAR_LENGTH
LOWER	LOWER
SUBSTR (second argument cannot be a negative number)	SUBSTRING
UPPER	UPPER

Group Functions

Oracle	Sybase
AVG	AVG
COUNT	COUNT
MAX	MAX
MIN	MIN
SUM	SUM

Other Functions

Oracle	Sybase
ABS	ABS
CEIL	CEILING
COS	COS
EXP	EXP
FLOOR	FLOOR
LN	LOG
LOG	LOG10
MOD	%
NOT NVL	IS NOT NULL
NVL	IS NULL
POWER	POWER
ROUND	ROUND
SIN	SIN
SQRT	SQRT
TAN	TAN

Data Dictionary

The Oracle Transparent Gateway for Sybase translates a query that refers to an Oracle database server data dictionary table into a query that retrieves the data from a Sybase data dictionary. You perform queries on data dictionary tables over the database link in the same way you query data dictionary tables in the Oracle database server. The gateway data dictionary is similar to the Oracle database server data dictionary in appearance and use.

This appendix contains the following sections:

- [Data Dictionary Support](#)
- [Data Dictionary Mapping](#)
- [Gateway Data Dictionary Descriptions](#)

Data Dictionary Support

Sybase System Tables

Sybase data dictionary information is stored in the Sybase database as Sybase system tables. All Sybase system tables have names prefaced with “sys”. The Sybase system tables define the structure of a database. When you change data definitions, Sybase reads and modifies the Sybase system tables to add information about the user tables.

Accessing the Gateway Data Dictionary

Accessing a gateway data dictionary table or view is identical to accessing a data dictionary in an Oracle database. You issue a SQL SELECT statement specifying a database link. The Oracle database server data dictionary view and column names are used to access the gateway data dictionary in an Oracle database. Synonyms of supported views are also acceptable. For example, the following statement queries the data dictionary table `ALL_TABLES` to retrieve all table names in the Sybase database:

```
SQL> SELECT * FROM "ALL_TABLES"@SYBS;
```

When a data dictionary access query is issued, the gateway:

1. Maps the requested table, view, or synonym to one or more Sybase system table names. The gateway translates all data dictionary column names to their corresponding Sybase column names within the query. If the mapping involves one Sybase system table, the gateway translates the requested table name to its corresponding Sybase system table name within the query. If the mapping involves multiple Sybase system tables, the gateway constructs a join in the query using the translated Sybase system table names.
2. Sends the translated query to Sybase.
3. Might convert the retrieved Sybase data to give it the appearance of the Oracle database server data dictionary table.
4. Passes the data dictionary information from the translated Sybase system table to the Oracle database server.

Note: The values returned when querying the gateway data dictionary might not be the same as the ones returned by the Oracle Enterprise Manager DESCRIBE command.

Direct Queries to Sybase Tables

Queries issued directly to individual Sybase system tables are allowed but they return different results because the Sybase system table column names differ from those of the data dictionary view. Also, certain columns in a Sybase system table cannot be used in data dictionary processing.

Supported Views and Tables

The gateway supports the following views and tables:

Table B–1 Gateway Supported Views and Tables

ALL_CATALOG	ALL_COL_COMMENTS
ALL_CONS_COLUMNS	ALL_CONSTRAINTS
ALL_INDEXES	ALL_OBJECTS
ALL_TAB_COLUMNS	ALL_TAB_COMMENTS
ALL_TABLES	ALL_USERS
ALL_VIEWS	DBA_CATALOG
DBA_COL_COMMENTS	DBA_OBJECTS
DBA_TABLES	DBA_TAB_COLUMNS
DBA_TAB_COMMENTS	DICT_COLUMNS
DICTIONARY	DUAL
TABLE_PRIVILEGES	USER_CATALOG
USER_COL_COMMENTS	USER_CONS_COLUMNS
USER_CONSTRAINTS	USER_INDEXES
USER_OBJECTS	USER_TAB_COLUMNS
USER_TAB_COMMENTS	USER_TABLES
USER_USER	USER_VIEWS

No other Oracle database server data dictionary tables or views are supported. If you use a view not on the list, you receive the Oracle database server error code for no more rows available.

Queries through the gateway of the supported data dictionary tables and views beginning with the characters ALL_ might return rows from the Sybase database

when you do not have access privileges for those Sybase objects. When querying an Oracle database with the Oracle data dictionary, rows are returned only for those objects you are permitted to access.

Data Dictionary Mapping

The tables in this section list Oracle data dictionary view names and the equivalent Sybase System tables used. A plus sign (+) indicates that a join operation is involved.

Table B-2 Oracle Data Dictionary View Names and Sybase Equivalents

View Name	Sybase System Table Name
ALL_CATALOG	sysusers + sysobjects
ALL_COL_COMMENTS	sysusers+sysobjects+syscolumns
ALL_CONS_COLUMNS	sp_pkeys + sp_fkeys
ALL_CONSTRAINTS	sp_pkeys + sp_fkeys
ALL_INDEXES	sysusers + sysindexes + sysobjects
ALL_OBJECTS	sysusers + sysobjects + sysindexes
ALL_TAB_COLUMNS	sysusers + sysobjects + syscolumns
ALL_TAB_COMMENTS	sysusers + sysobjects
ALL_TABLES	sysusers + sysobjects
ALL_USERS	sysusers
ALL_VIEWS	sysusers + sysobjects + syscomments
DBA_CATALOG	sysusers + sysobjects
DBA_COL_COMMENTS	sysusers + sysobjects + syscolumns
DBA_OBJECTS	sysusers + sysobjects + sysindexes
DBA_TABLES	sysusers + sysobjects
DBA_TAB_COLUMNS	sysusers + sysobjects + syscolumns
DBA_TAB_COMMENTS	sysusers + sysobjects
DICT_COLUMNS	sysobjects + syscolumns
DICTIONARY	sysobjects
DUAL	sysusers

Table B–2 (Cont.) Oracle Data Dictionary View Names and Sybase Equivalents

View Name	Sybase System Table Name
TABLE_PRIVILEGES	sysprotects + sysusers + sysobjects
USER_CATALOG	sysusers + sysobjects
USER_COL_COMMENTS	sysusers + sysobjects + syscolumns
USER_CONS_COLUMNS	sp_pkeys + sp_fkeys
USER_CONSTRAINTS	sp_pkeys + sp_fkeys
USER_INDEXES	sysusers + sysindexes + sysobjects
USER_OBJECTS	sysusers + sysobjects + sysindexes
USER_TAB_COLUMNS	sysusers + sysobjects + syscolumns
USER_TAB_COMMENTS	sysusers + sysobjects
USER_TABLES	sysusers + sysobjects
USER_USERS	sysusers
USER_VIEWS	sysusers + sysobjects + syscomments

Default Column Values

There is a minor difference between the gateway data dictionary and a typical Oracle database server data dictionary. The Oracle database server columns that are missing in a Sybase data dictionary table are filled with zeros, spaces, null values, not-applicable values (N.A.), or default values, depending on the column type.

Gateway Data Dictionary Descriptions

The gateway data dictionary tables and views provide the following information:

- Name, datatype, and width of each column
- The contents of columns with fixed values

They are described here with information retrieved by an Oracle Enterprise Manager DESCRIBE command. The values in the Null? column might differ from the Oracle database server data dictionary tables and views. Any default value is shown to the right of an item, but this is not information returned by DESCRIBE.

Table B-3 ALL_CATALOG

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(5)	"TABLE" or "VIEW"

Table B-4 ALL_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
COLUMN_NAME	NOT NULL	VARCHAR2(30)	
COMMENTS	NOT NULL	CHAR(1)	

Table B-5 ALL_CONS_COLUMNS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
CONSTRAINT_NAME		VARCHAR2(30)	
TABLE_NAME		VARCHAR2(30)	
COLUMN_NAME		VARCHAR2(4000)	
POSITION		NUMBER	

Table B-6 ALL_CONSTRAINTS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
CONSTRAINT_NAME		VARCHAR2(30)	
CONSTRAINT_TYPE		VARCHAR2(1)	"R" or "P"
TABLE_NAME		VARCHAR2(30)	
SEARCH_CONDITION		VARCHAR2(1)	NULL
R_OWNER		VARCHAR2(30)	

Table B-6 (Cont.) ALL_CONSTRAINTS

Name	Null?	Type	Value
R_CONSTRAINT_NAME		VARCHAR2(30)	
DELETE_RULE		VARCHAR2(9)	"CASCADE" or "NO ACTION" or "SET NULL"
STATUS		VARCHAR2(1)	NULL
DEFERRABLE		VARCHAR2(1)	NULL
DEFERRED		VARCHAR2(1)	NULL
VALIDATED		VARCHAR2(1)	NULL
GENERATED		VARCHAR2(1)	NULL
BAD		VARCHAR2(1)	NULL
RELY		VARCHAR2(1)	NULL
LAST_CHANGE		DATE	NULL

Table B-7 ALL_INDEXES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
INDEX_NAME		VARCHAR2(30)	
INDEX_TYPE		VARCHAR2(1)	NULL
TABLE_OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR(7)	"TABLE" or "CLUSTER"
UNIQUENESS		VARCHAR2(1)	NULL
COMPRESSION		VARCHAR2(1)	NULL
PREFIX_LENGTH		NUMBER	0
TABLESPACE_NAME		VARCHAR2(1)	NULL
INI_TRANS		NUMBER	0
MAX_TRANS		NUMBER	0

Table B-7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
INITIAL_EXTENT		NUMBER	0
NEXT_EXTENT		NUMBER	0
MIN_EXTENTS		NUMBER	0
MAX_EXTENTS		NUMBER	0
PCT_INCREASE		NUMBER	0
PCT_THRESHOLD		NUMBER	0
INCLUDE_COLUMN		NUMBER	0
FREELISTS		NUMBER	0
FREELIST_GROUPS		NUMBER	0
PCT_FREE		NUMBER	0
LOGGING		VARCHAR2(1)	NULL
BLEVEL		NUMBER	0
LEAF_BLOCKS		NUMBER	0
DISTINCT_KEYS		NUMBER	0
AVG_LEAF_BLOCKS_PER_KEY		NUMBER	0
AVG_DATA_BLOCKS_PER_KEY		NUMBER	0
CLUSTERING_FACTOR		NUMBER	0
STATUS		VARCHAR2(1)	NULL
NUM_ROWS		NUMBER	0
SAMPLE_SIZE		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
DEGREE		VARCHAR2(1)	NULL
INSTANCES		VARCHAR2(1)	NULL
PARTITIONED		VARCHAR2(1)	NULL
TEMPORARY		VARCHAR2(1)	NULL
GENERATED		VARCHAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL

Table B-7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
BUFFER_POOL		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL
DURATION		VARCHAR2(1)	NULL
PCT_DIRECT_ACCESS		NUMBER	0
ITYP_OWNER		VARCHAR2(1)	NULL
ITYP_NAME		VARCHAR2(1)	NULL
PARAMETERS		VARCHAR2(1)	NULL
GLOBAL_STATS		VARCHAR2(1)	NULL
DOMIDX_STATUS		VARCHAR2(1)	NULL
DOMIDX_OPSTATUS		VARCHAR2(1)	NULL
FUNCIDX_STATUS		VARCHAR2(1)	NULL

Table B-8 ALL_OBJECTS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
OBJECT_NAME		VARCHAR2(30)	
SUBOBJECT_NAME		VARCHAR2(1)	NULL
OBJECT_ID		NUMBER	
DATA_OBJECT_ID		NUMBER	0
OBJECT_TYPE		VARCHAR2(9)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED		DATE	
LAST_DDL_TIME		DATE	
TIMESTAMP		VARCHAR2(1)	NULL
STATUS		VARCHAR2(5)	"VALID"
TEMPORARY		VARCHAR2(1)	NULL

Table B-8 (Cont.) ALL_OBJECTS

Name	Null?	Type	Value
GENERATED		VARCHAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL

Table B-9 ALL_TAB_COLUMNS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
TABLE_NAME		VARCHAR2(30)	
COLUMN_NAME		VARCHAR2(30)	
DATA_TYPE		VARCHAR2(8)	
DATA_TYPE_MOD		VARCHAR2(1)	NULL
DATA_TYPE_OWNER		VARCHAR2(1)	NULL
DATA_LENGTH		NUMBER	
DATA_PRECISION		NUMBER	
DATA_SCALE		NUMBER	
NULLABLE		VARCHAR2(1)	NULL
COLUMN_ID		NUMBER	
DEFAULT_LENGTH		NUMBER	0
DATA_DEFAULT		VARCHAR2(1)	NULL
NUM_DISTINCT		NUMBER	0
LOW_VALUE		NUMBER	0
HIGH_VALUE		NUMBER	0
DENSITY		NUMBER	0
NUM_NULLS		NUMBER	0
NUM_BUCKETS		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
SAMPLE_SIZE		NUMBER	0
CHARACTER_SET_NAME		VARCHAR2(1)	NULL

Table B–9 (Cont.) ALL_TAB_COLUMNS

Name	Null?	Type	Value
CHAR_COL_DEC_LENGTH		NUMBER	0
GLOBAL_STATS		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL
AVG_COL_LEN		NUMBER	0

Table B–10 ALL_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS		VARCHAR2(1)	NULL

Table B–11 ALL_TABLES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLESPACE_NAME		VARCHAR2(1)	NULL
CLUSTER_NAME		VARCHAR2(1)	NULL
IOT_NAME		VARCHAR2(1)	NULL
PCT_FREE		NUMBER	0
PCT_USED		NUMBER	0
INI_TRANS		NUMBER	0
MAX_TRANS		NUMBER	0
INITIAL_EXTENT		NUMBER	0
NEXT_EXTENT		NUMBER	0
MIN_EXTENTS		NUMBER	0
MAX_EXTENTS		NUMBER	0

Table B-11 (Cont.) ALL_TABLES

Name	Null?	Type	Value
PCT_INCREASE		NUMBER	0
FREELISTS		NUMBER	0
FREELIST_GROUPS		NUMBER	0
LOGGING		VARCHAR2(1)	NULL
BACKED_UP		VARCHAR2(1)	NULL
NUM_ROWS		NUMBER	0
BLOCKS		NUMBER	0
EMPTY_BLOCKS		NUMBER	0
AVG_SPACE		NUMBER	0
CHAIN_CNT		NUMBER	0
AVG_ROW_LEN		NUMBER	0
AVG_SPACE_FREELIST_BLOCKS		NUMBER	0
NUM_FREELIST_BLOCKS		NUMBER	0
DEGREE		VARCHAR2(1)	NULL
INSTANCES		VARCHAR2(1)	NULL
CACHE		VARCHAR2(1)	NULL
TABLE_LOCK		VARCHAR2(1)	NULL
SAMPLE_SIZE		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
PARTITIONED		VARCHAR2(1)	NULL
IOT_TYPE		VARCHAR2(1)	NULL
TEMPORARY		VARHCAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL
NESTED		VARCHAR2(1)	NULL
BUFFER_POOL		VARCHAR2(1)	NULL
ROW_MOVEMENT		VARCHAR2(1)	NULL
GLOBAL_STATS		VARCHAR2(1)	NULL

Table B–11 (Cont.) ALL_TABLES

Name	Null?	Type	Value
USER_STATS		VARCHAR2(1)	NULL
DURATION		VARHCAR2(1)	NULL
SKIP_CORRUPT		VARCHAR2(1)	NULL
MONITORING		VARCHAR2(1)	NULL

Table B–12 ALL_USERS

Name	Null?	Type	Value
USERNAME	NOT NULL	VARCHAR2(30)	
USER_ID	NOT NULL	NUMBER(5)	
CREATED		VARCHAR2(0)	NULL

Table B–13 ALL_VIEWS

Name	Null?	Type	Value
OWNER	NOT NULL	CHAR(30)	
VIEW_NAME	NOT NULL	CHAR(30)	
TEXT_LENGTH	NOT NULL	NUMBER(10)	0
TEXT	NOT NULL	VARCHAR2(255)	
TYPE_TEXT_LENGTH	NOT NULL	NUMBER(10)	0
TYPE_TEXT	NOT NULL	CHAR(1)	
OID_TEXT_LENGTH	NOT NULL	NUMBER(10)	0
OID_TEXT	NOT NULL	CHAR(1)	
VIEW_TYPE_OWNER	NOT NULL	CHAR(1)	
VIEW_TYPE	NOT NULL	CHAR(1)	

Table B–14 DBA_CATALOG

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	

Table B-14 (Cont.) DBA_CATALOG

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(5)	"TABLE" or "VIEW"

Table B-15 DBA_COL_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
COLUMN_NAME	NOT NULL	VARCHAR2(30)	
COMMENTS	NOT NULL	CHAR(1)	NULL

Table B-16 DBA_OBJECTS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
OBJECT_NAME		VARCHAR2(30)	
SUBOBJECT_NAME		VARCHAR2(1)	NULL
OBJECT_ID		NUMBER	
DATA_OBJECT_ID		NUMBER	0
OBJECT_TYPE		VARCHAR2(9)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED		DATE	
LAST_DDL_TIME		DATE	
TIMESTAMP		VARCHAR2(1)	NULL
STATUS		VARCHAR2(5)	NULL
TEMPORARY		VARCHAR2(1)	NULL
GENERATED		VARCHAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL

Table B-17 DBA_TAB_COLUMNS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
TABLE_NAME		VARCHAR2(30)	
COLUMN_NAME		VARCHAR2(30)	
DATA_TYPE		VARCHAR2(8)	
DATA_TYPE_MOD		VARCHAR2(1)	NULL
DATA_TYPE_OWNER		VARCHAR2(1)	NULL
DATA_LENGTH		NUMBER	
DATA_PRECISION		NUMBER	
DATA_SCALE		NUMBER	
NULLABLE		VARCHAR2(1)	NULL
COLUMN_ID		NUMBER	
DEFAULT_LENGTH		NUMBER	0
DATA_DEFAULT		VARCHAR2(1)	NULL
NUM_DISTINCT		NUMBER	0
LOW_VALUE		NUMBER	0
HIGH_VALUE		NUMBER	0
DENSITY		NUMBER	0
NUM_NULLS		NUMBER	0
NUM_BUCKETS		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
SAMPLE_SIZE		NUMBER	0
CHARACTER_SET_NAME		VARCHAR2(1)	NULL
CHAR_COL_DEC_LENGTH		NUMBER	0
GLOBAL_STATS		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL
AVG_COL_LEN		NUMBER	0

Table B-18 DBA_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS		VARCHAR2(1)	NULL

Table B-19 DBA_TABLES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLESPACE_NAME		VARCHAR2(1)	NULL
CLUSTER_NAME		VARCHAR2(1)	NULL
IOT_NAME		VARCHAR2(1)	NULL
PCT_FREE		NUMBER	0
PCT_USED		NUMBER	0
INI_TRANS		NUMBER	0
MAX_TRANS		NUMBER	0
INITIAL_EXTENT		NUMBER	0
NEXT_EXTENT		NUMBER	0
MIN_EXTENTS		NUMBER	0
MAX_EXTENTS		NUMBER	0
PCT_INCREASE		NUMBER	0
FREELISTS		NUMBER	0
FREELIST_GROUPS		NUMBER	0
LOGGING		VARCHAR2(1)	NULL
BACKED_UP		VARCHAR2(1)	NULL
NUM_ROWS		NUMBER	0

Table B-19 (Cont.) DBA_TABLES

Name	Null?	Type	Value
BLOCKS		NUMBER	0
EMPTY_BLOCKS		NUMBER	0
AVG_SPACE		NUMBER	0
CHAIN_CNT		NUMBER	0
AVG_ROW_LEN		NUMBER	0
AVG_SPACE_FREELIST_BLOCKS		NUMBER	0
NUM_FREELIST_BLOCKS		NUMBER	0
DEGREE		VARCHAR2(1)	NULL
INSTANCES		VARCHAR2(1)	NULL
CACHE		VARCHAR2(1)	NULL
TABLE_LOCK		VARCHAR2(1)	NULL
SAMPLE_SIZE		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
PARTITIONED		VARCHAR2(1)	NULL
IOT_TYPE		VARCHAR2(1)	NULL
TEMPORARY		VARHCAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL
NESTED		VARCHAR2(1)	NULL
BUFFER_POOL		VARCHAR2(1)	NULL
ROW_MOVEMENT		VARCHAR2(1)	NULL
GLOBAL_STATS		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL
DURATION		VARHCAR2(1)	NULL
SKIP_CORRUPT		VARCHAR2(1)	NULL
MONITORING		VARCHAR2(1)	NULL

Table B–20 *DICT_COLUMNS*

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
COLUMN_NAME	NOT NULL	VARCHAR2(30)	
COMMENTS	NOT NULL	CHAR2(1)	NULL

Table B–21 *DICTIONARY*

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
COMMENTS	NOT NULL	CHAR(1)	

Table B–22 *DUAL*

Name	Null?	Type	Value
DUMMY	NOT NULL	CHAR(1)	"X"

Table B–23 *TABLE_PRIVILEGES*

Name	Null?	Type	Value
GRANTEE	NOT NULL	VARCHAR2(30)	
OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
GRANTOR	NOT NULL	VARCHAR2(30)	
SELECT_PRIV		VARCHAR2(1)	"Y"
INSERT_PRIV		VARCHAR2(1)	"A"
DELETE_PRIV		VARCHAR2(1)	"Y"
UPDATE_PRIV		VARCHAR2(1)	"A"
REFERENCES_PRIV		VARCHAR2(1)	"A"
ALTER_PRIV		VARCHAR2(1)	"Y"
INDEX_PRIV		VARCHAR2(1)	"Y"
CREATED		VARCHAR2(0)	NULL

Table B–24 USER_CATALOG

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(5)	"TABLE" or "VIEW"

Table B–25 USER_COL_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
COLUMN_NAME	NOT NULL	VARCHAR2(30)	
COMMENTS		VARCHAR2(1)	NULL

Table B–26 USER_CONS_COLUMNS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
CONSTRAINT_NAME		VARCHAR2(30)	
TABLE_NAME		VARCHAR2(30)	
COLUMN_NAME		VARCHAR2(4000)	
POSITION		NUMBER	

Table B–27 USER_CONSTRAINTS

Name	Null?	Type	Value
OWNER		VARCHAR2(30)	
CONSTRAINT_NAME		VARCHAR2(30)	
CONSTRAINT_TYPE		VARCHAR2(1)	"R" or "P"
TABLE_NAME		VARCHAR2(30)	
SEARCH_CONDITION		VARCHAR2(1)	NULL
R_OWNER		VARCHAR2(30)	
R_CONSTRAINT_NAME		VARCHAR2(30)	

Table B–27 (Cont.) USER_CONSTRAINTS

Name	Null?	Type	Value
DELETE_RULE		VARCHAR2(9)	"CASCADE" or "NOACTION" or "SET NULL"
STATUS		VARCHAR2(1)	NULL
DEFERRABLE		VARCHAR2(1)	NULL
DEFERRED		VARCHAR2(1)	NULL
VALIDATED		VARCHAR2(1)	NULL
GENERATED		VARCHAR2(1)	NULL
BAD		VARCHAR2(1)	NULL
RELY		VARCHAR2(1)	NULL
LAST_CHANGE		DATE	NULL

Table B–28 USER_INDEXES

Name	Null?	Type	Value
INDEX_NAME		VARCHAR2(30)	
INDEX_TYPE		VARCHAR2(1)	NULL
TABLE_OWNER	NOT NULL	VARCHAR2(30)	
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(7)	"TABLE" or "CLUSTER"
UNIQUENESS		VARCHAR2(1)	NULL
COMPRESSION		VARCHAR2(1)	NULL
PREFIX_LENGTH		NUMBER	0
TABLESPACE_NAME		VARCHAR2(1)	NULL
INI_TRANS		NUMBER	0
MAX_TRANS		NUMBER	0
INITIAL_EXTENT		NUMBER	0
NEXT_EXTENT		NUMBER	0

Table B–28 (Cont.) USER_INDEXES

Name	Null?	Type	Value
MIN_EXTENTS		NUMBER	0
MAX_EXTENTS		NUMBER	0
PCT_INCREASE		NUMBER	0
PCT_THRESHOLD		NUMBER	0
INCLUDE_COLUMN		NUMBER	0
FREELISTS		NUMBER	0
FREELIST_GROUPS		NUMBER	0
PCT_FREE		NUMBER	0
LOGGING		VARCHAR2(1)	NULL
BLEVEL		NUMBER	0
LEAF_BLOCKS		NUMBER	0
DISTINCT_KEYS		NUMBER	0
AVG_LEAF_BLOCKS_PER_KEY		NUMBER	0
AVG_DATA_BLOCKS_PER_KEY		NUMBER	0
CLUSTERING_FACTOR		NUMBER	0
STATUS		VARCHAR2(1)	NULL
NUM_ROWS		NUMBER	0
SAMPLE_SIZE		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
DEGREE		VARCHAR2(1)	NULL
INSTANCES		VARCHAR2(1)	NULL
PARTITIONED		VARCHAR2(1)	NULL
TEMPORARY		VARCHAR2(1)	NULL
GENERATED		VARCHAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL
BUFFER_POOL		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL

Table B–28 (Cont.) USER_INDEXES

Name	Null?	Type	Value
DURATION		VARHCAR2(1)	NULL
PCT_DIRECT_ACCESS		NUMBER	0
ITYP_OWNER		VARCHAR2(1)	NULL
ITYP_NAME		VARCHAR2(1)	NULL
PARAMETERS		VARCHAR2(1)	NULL
GLOBAL_STATS		VARCHAR2(1)	NULL
DOMIDX_STATUS		VARCHAR2(1)	NULL
DOMIDX_OPSTATUS		VARCHAR2(1)	NULL
FUNCIDX_STATUS		VARCHAR2(1)	NULL

Table B–29 USER_OBJECTS

Name	Null?	Type	Value
OBJECT_NAME		VARCHAR2(30)	
SUBOBJECT_NAME		VARCHAR2(1)	NULL
OBJECT_ID		NUMBER	
DATA_OBJECT_ID		NUMBER	0
OBJECT_TYPE		VARCHAR2(9)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED		DATE	
LAST_DDL_TIME		DATE	
TIMESTAMP		VARCHAR2(1)	NULL
STATUS		VARCHAR2(5)	"VALID"
TEMPORARY		VARCHAR2(1)	NULL
GENERATED		VARCHAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL

Table B-30 USER_TAB_COLUMNS

Name	Null?	Type	Value
TABLE_NAME		VARCHAR2(30)	
COLUMN_NAME		VARCHAR2(30)	
DATA_TYPE		VARCHAR2(8)	
DATA_TYPE_MOD		VARCHAR2(1)	NULL
DATA_TYPE_OWNER		VARCHAR2(1)	NULL
DATA_LENGTH		NUMBER	
DATA_PRECISION		NUMBER	
DATA_SCALE		NUMBER	
NULLABLE		VARCHAR2(1)	"Y" or "N"
COLUMN_ID		NUMBER	
DEFAULT_LENGTH		NUMBER	0
DATA_DEFAULT		VARCHAR2(1)	NULL
NUM_DISTINCT		NUMBER	0
LOW_VALUE		NUMBER	0
HIGH_VALUE		NUMBER	0
DENSITY		NUMBER	0
NUM_NULLS		NUMBER	0
NUM_BUCKETS		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
SAMPLE_SIZE		NUMBER	0
CHARACTER_SET_NAME		VARCHAR2(1)	NULL
CHAR_COL_DECL_LENGTH		NUMBER	0
GLOBAL_STATS		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL
AVG_COL_LEN		NUMBER	0

Table B–31 USER_TAB_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLE_TYPE		VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS		VARCHAR2(1)	NULL

Table B–32 USER_TABLES

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	
TABLESPACE_NAME		VARCHAR2(1)	NULL
CLUSTER_NAME		VARCHAR2(1)	NULL
IOT_NAME		VARCHAR2(1)	NULL
PCT_FREE		NUMBER	0
PCT_USED		NUMBER	0
INI_TRANS		NUMBER	0
MAX_TRANS		NUMBER	0
INITIAL_EXTENT		NUMBER	0
NEXT_EXTENT		NUMBER	0
MIN_EXTENTS		NUMBER	0
MAX_EXTENTS		NUMBER	0
PCT_INCREASE		NUMBER	0
FREELISTS		NUMBER	0
FREELIST_GROUPS		NUMBER	0
LOGGING		VARCHAR2(1)	NULL
BACKED_UP		VARCHAR2(1)	NULL
NUM_ROWS		NUMBER	0
BLOCKS		NUMBER	0
EMPTY_BLOCKS		NUMBER	0

Table B-32 (Cont.) USER_TABLES

Name	Null?	Type	Value
AVG_SPACE		NUMBER	0
CHAIN_CNT		NUMBER	0
AVG_ROW_LEN		NUMBER	0
AVG_SPACE_FREELIST_BLOCKS		NUMBER	0
NUM_FREELIST_BLOCKS		NUMBER	0
DEGREE		VARCHAR2(1)	NULL
INSTANCES		VARCHAR2(1)	NULL
CACHE		VARCHAR2(1)	NULL
TABLE_LOCK		VARCHAR2(1)	NULL
SAMPLE_SIZE		NUMBER	0
LAST_ANALYZED		VARCHAR2(0)	NULL
PARTITIONED		VARCHAR2(1)	NULL
IOT_TYPE		VARCHAR2(1)	NULL
TEMPORARY		VARHCAR2(1)	NULL
SECONDARY		VARCHAR2(1)	NULL
NESTED		VARCHAR2(1)	NULL
BUFFER_POOL		VARCHAR2(1)	NULL
ROW_MOVEMENT		VARCHAR2(1)	NULL
GLOBAL_STATS		VARCHAR2(1)	NULL
USER_STATS		VARCHAR2(1)	NULL
DURATION		VARCHAR2(1)	NULL
SKIP_CORRUPT		VARCHAR2(1)	NULL
MONITORING		VARCHAR2(1)	NULL

Table B-33 USER_USERS

Name	Null?	Type	Value
USERNAME	NOT NULL	VARCHAR2(30)	

Table B-33 (Cont.) USER_USERS

Name	Null?	Type	Value
USER_ID	NOT NULL	NUMBER(5)	
ACCOUNT_STATUS		VARCHAR2(4)	"OPEN"
LOCK_DATE		VARCHAR2(0)	NULL
EXPIRY_DATE		VARCHAR2(0)	NULL
DEFAULT_TABLESPACE		VARCHAR2(1)	NULL
TEMPORARY_TABLESPACE		VARCHAR2(1)	NULL
CREATED		VARCHAR2(0)	NULL
INITIAL_RSRC_CONSUMER_GROUP		VARCHAR2(1)	NULL
EXTERNAL_NAME		VARCHAR2(1)	NULL

Table B-34 USER_VIEWS

Name	Null?	Type	Value
VIEW_NAME	NOT NULL	VARCHAR2(30)	
TEXT_LENGTH		NUMBER	0
TEXT	NOT NULL	VARCHAR2(255)	
TYPE_TEXT_LENGTH		NUMBER	0
TYPE_TEXT		VARCHAR2(1)	NULL
OID_TEXT_LENGTH		NUMBER	0
OID_TEXT		VARCHAR2(1)	NULL
VIEW_TYPE_OWNER		VARCHAR2(1)	NULL
VIEW_TYPE		VARCHAR2(1)	NULL

Heterogeneous Services Initialization Parameters

The Oracle database server `init.ora` parameters are distinct from heterogeneous services (HS) initialization parameters. Set Heterogeneous Services parameters in the initialization parameter file using an agent-specific mechanism, or set them in the Oracle data dictionary using the `DBMS_HS` package. All string values for Heterogeneous Services parameters must be lowercase.

This appendix contains information about the following Heterogeneous Services initialization parameters:

- [HS_FDS_CONNECT_INFO](#)
- [HS_FDS_DEFAULT_OWNER](#)
- [HS_FDS_PROC_IS_FUNC](#)
- [HS_FDS_RECOVERY_ACCOUNT](#)
- [HS_FDS_RECOVERY_PWD](#)
- [HS_FDS_TRACE_LEVEL](#)
- [HS_FDS_TRANSACTION_MODEL](#)

See Also: *Oracle 8i Distributed Database Systems* for information on other available initialization parameters.

Initialization Parameter File Syntax

The syntax for the initialization parameter file is as follows:

1. The file is a sequence of commands.
2. Each command should start on a separate line.
3. End of line is considered a command terminator (unless escaped with a backslash).
4. Each command can have one of the following forms:
 - a. `<param> = <value>`
 - b. `set <param> = <value>`
 - c. `private <param> = <value>`
 - d. `set private <param> = <value>`

where:

`<param>` is an initialization parameter name.

`<value>` is the initialization parameter value.

'set' and 'private' are keywords.

5. 'set' and 'private' are reserved keywords. You cannot use either as an initialization parameter name. The 'set' keyword indicates that the initialization parameter should be set as an environment variable in the agent. The 'private' keyword indicates that the initialization parameter should be private to the agent and should not be uploaded to the server. Most initialization parameters should not be private. If, however, you are storing something sensitive like a password in the initialization parameter file, then you may not want it uploaded to the server because the initialization parameters and values are not encrypted when uploaded. Making these initialization parameters private prevents the upload from happening.
6. An initialization parameter name is a string of characters starting with a letter and consisting of letters, digits and underscores. Initialization parameter names are case sensitive.
7. An initialization parameter value is either:
 - a. A string of characters that does not contain any backslashes, white space or double quotation marks ("")

- b. A quoted string beginning with a double quotation mark and ending with a double quotation mark. The following can be used inside a quoted string:

- * backslash (\) is the escape character
- * \n inserts a newline
- * \t inserts a tab
- * \" inserts a double quotation mark
- * \\ inserts a backslash

A backslash at the end of the line continues the string on the next line. If a backslash precedes any other character then the backslash is ignored.

If there is a syntax error in an initialization parameter file, none of the settings take effect.

HS_FDS_CONNECT_INFO

Default value:	none
Range of values:	not applicable

Specifies the information needed to connect to the Sybase database.

This is a required parameter, whose format is:

HS_FDS_CONNECT_INFO=server_name.database_name

where:

- *server_name* is the name of the server machine for the Sybase data and *database_name* is the name of the database.

The entries for both the *server_name* and *database_name* are case-sensitive.

HS_FDS_DEFAULT_OWNER

Default value:	none
Range of values:	any value

The name of the table owner that is used for the Sybase tables if an owner is not specified in the SQL statements.

Note: If this parameter is not specified and the owner is not explicitly specified in the SQL statement, then the user name of the Oracle user or the user name specified when creating the database link is used.

HS_FDS_PROC_IS_FUNC

Default value:	none
Range of values:	any value

Enables return values from functions. By default, all stored procedures and functions do not return a return value to the user.

Note: If you set this initialization parameter, you must change the syntax of the procedure execute statement for all existing stored procedures.

HS_FDS_RECOVERY_ACCOUNT

Default value:	RECOVER
Range of values:	any valid userid

Specifies the name of the recovery account used for the commit-confirm transaction model. An account with user name and password must be set up at Sybase. For more information about the commit-confirm model, see the HS_FDS_TRANSACTION_MODEL parameter.

HS_FDS_RECOVERY_PWD

Default value:	RECOVER
Range of values:	any valid password

Specifies the password of the recovery account used for the commit-confirm transaction model set up at Sybase. For more information about the commit-confirm model, see the HS_FDS_TRANSACTION_MODEL parameter.

HS_FDS_TRACE_LEVEL

Default value:	off
Range of values:	off, on

Specifies whether error tracing is turned on or off for gateway connectivity.

The following values are valid:

- OFF disables the tracing of error messages.
- ON enables the tracing of error messages that occur when you encounter problems. The results are written to a gateway connectivity log file, in `$ORACLE_HOME/TG4SYBS/LOG`.

HS_FDS_TRANSACTION_MODEL

Default value:	COMMIT_CONFIRM
Range of values:	COMMIT_CONFIRM, READ_ONLY, or SINGLE_SITE

Specifies the type of transaction model that is used when the Sybase database is updated by a transaction.

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the Sybase database and allows the gateway to be part of a distributed update. To use the commit-confirm model, the following items must be created in the Sybase database:
 - Transaction log table. The table name is HS_TRANSACTION_LOG. The transaction log table must be granted SELECT, DELETE, and INSERT privileges set to public.
 - Recovery account. The account name is assigned with the HS_FDS_RECOVERY_ACCOUNT parameter.
 - Recovery account password. The password is assigned with the HS_FDS_RECOVERY_PWD parameter.
- READ_ONLY provides read access to the Sybase database.
- SINGLE_SITE provides read and write access to the Sybase database. However, the gateway cannot participate in distributed updates.

Index

Symbols

||, 1-17

A

ALTER statement, A-2

Arithmetic operators, A-4

B

BINARY data type, 5-4

BIT data type, 5-4

C

Case rules, 1-6

Case studies, 6-1

CD-ROM drive requirements, 1-3

Chained mode, 1-5

CHAR data type, 5-4

COMMIT

restrictions, 1-11

Commit point site, 1-11, 5-8

Comparison operators, A-4

CONCAT operator, 1-17

Configuring

two-phase commit, 4-9

Configuring the gateway, 4-2

CONNECT BY clause, 1-12

COPY command, 1-13

Copying data

COPY command, 5-9

from Oracle database server to SQL Server, 5-9

from SQL Server to Oracle database server, 5-10

INSERT statement, 5-10

CREATE statement, A-2

CREATE TABLE command, 5-10

Creating

transaction log table, 4-10

Cursor loops

restrictions, 1-11

D

Data definition language, A-2

Data dictionary

views, B-3

Data type

BINARY, 5-4

BIT, 5-4

CHAR, 5-4

conversion, 1-8

DATE, 5-4

DATETIME, 5-4

DECIMAL, 5-4

DOUBLE PRECISION, 5-4

FLOAT, 5-4

FLOAT data type, 1-15

IMAGE, 5-4

INTEGER, 5-4

LONG, 5-4

LONG RAW, 5-4

MONEY, 5-4

NCHAR, 5-4

NUMBER, 5-4

NUMERIC, 5-4

NVARCHAR, 5-4

RAW, 5-4
REAL, 5-4
SMALL DATETIME, 5-4
SMALL MONEY, 5-4
SMALLINT, 5-4
TEXT, 5-4
TINYINT, 5-4
VARBINARY, 1-15, 5-4
VARCHAR, 5-4
VARCHAR2, 5-4
Database link
 behavior, 4-11
DATE data type, 5-4
DATETIME data type, 5-4
DDL statement, 1-11
DECIMAL data type, 5-4
DELETE statement, 6-9, A-2, A-3
Demo build SQL script, 6-2
Demonstration files, 6-2
Demonstration tables, 6-3
Demonstration tables build SQL script, 6-3
Disk space requirements, 1-3
Distributed queries, 5-7
DOUBLE PRECISION data type, 5-4
DROP statement, A-2

E

Encrypted format login, 1-15
Error message conventions, 7-6
Error messages
 error tracing, C-5
 ORA-12203, 7-4
 ORA-12500, 7-5
 Trace output, 7-9
Errors
 ORA-02070, 1-11

F

FLOAT data type, 5-4
Functions in SQL, 5-6

G

Gateway
 case studies, 6-1
 components, 2-3
 data dictionary tables, B-2
 default SID, 4-2
 distributed query, 5-7
 error messages, 7-7
 how it works, 2-5
 installing, 3-1
 pass-through feature, 5-5
 remote data access, 2-7
 supported functions, A-1
 supported SQL syntax, A-2
 system identifier (SID), 4-2
 two-phase commit, 2-8, 4-9, 5-8
GRANT statement, A-2
Group functions, A-5

H

Hardware requirements, 1-3
Hexadecimal notation, 1-7
HS_COMMIT_POINT_STRENGTH setting, 5-8
HS_FDS_CONNECT_INFO initialization
 parameter, C-3
HS_FDS_DEFAULT_OWNER initialization
 parameter, 4-11, C-4
HS_FDS_PROC_IS_FUNC initialization
 parameter, C-4
HS_FDS_RECOVERY_ACCOUNT initialization
 parameter, C-4
HS_FDS_RECOVERY_PWD initialization
 parameter, C-5
HS_FDS_TRACE_LEVEL initialization
 parameter, C-5
HS_FDS_TRANSACTION_MODEL initialization
 parameter, C-6
HS_TRANSACTION_LOG, 4-10
HS_TRANSACTION_LOG table, 5-8

I

IMAGE data type, 5-4
Informix software requirements, 1-4

Initialization parameter file
 customizing, 4-2
initsid.ora file, 4-2
INSERT statement, 1-13, 6-11, A-2, A-3
Installing the gateway, 3-1 to 3-8
INTEGER data type, 5-4

K

Known restrictions, 1-9

L

listener.ora file, 4-14
 example, 4-4
 location, 4-3
Locking, database, 1-9
LONG data type, 5-4
LONG RAW data type, 5-4

M

Memory requirements, 1-3
MONEY data type, 5-4

N

NCHAR data type, 5-4
Net8
 configuration problems, 7-3
 configuring, 4-3
 requirements, 1-4
NLS
 about, 1-14
NULL values, 1-5
NUMBER data type, 5-4
NUMERIC data type, 5-4
NVARCHAR data type, 5-4
NVL function, 6-7

O

Objects, naming rules, 1-6
Operating system requirements, 1-3
ORA-02070, 1-11
ORA-12154, 7-3

ORA-12203, 7-4
ORA-12500, 7-5
Oracle database server
 description, 2-4
 error messages, 7-7
 requirements, 1-3
 SQL construct processing, 5-3

P

Passing commands to database, 1-11
Pass-through feature, 1-11
PL/SQL, 1-17
Processor requirements, 1-3
Product set for Informix gateway, 1-2

R

RAW data type, 5-4
REAL data type, 5-4
RECOVERY_ACCOUNT
 account username, 4-9
 creating a recovery account, 4-9
Retrieving data from SQL Server, 5-2
ROLLBACK
 restrictions, 1-11
ROWID, 1-12, 1-13

S

savepoint support, 1-11
SELECT statement, 5-2, A-2, B-2
SID, 4-2
SMALL DATETIME data type, 5-4
SMALLINT data type, 5-4
Software requirements, 1-3
Specifying an Owner, 4-11
Stored procedures, 1-17
 running in chained mode, 1-5
Stored procedures in SQL Server, 5-6
String functions, A-4
SUM function, 6-7
Synonyms, 5-5

T

TEXT data type, 5-4

TINYINT data type, 5-4

TNS listener

- configuring for gateway, 4-3

- starting, 4-5

- starting for multiple gateway instances, 4-15

Trace output, 7-9

Transaction log table

- creating, 4-10

Transaction modes, 1-5

transactional capability, 1-11

transactional integrity, 1-10

Troubleshooting

- Net8 configuration problems, 7-3

TRUNCATE statement, A-2

Two-phase commit, 1-11, 2-8, 5-8

- configuration, 4-9

- transaction log table, 4-10

U

UPDATE statement, 1-12, 5-3, 6-10, 6-13, 6-14, A-3

Updating SQL Server data, 5-3

V

VARBINARY data type, 1-15, 5-4

VARCHAR data type, 5-4

VARCHAR2 data type, 5-4

W

WHERE CURRENT OF clause, 1-12