Oracle® Transparent Gateway for Microsoft SQL Server

Administrator's Guide

10*g* Release 2 (10.2) for Microsoft Windows (32-bit)

B14270-01

June 2005

Oracle Transparent Gateway for Microsoft SQL Server Administrator's Guide, 10*g* Release 2 (10.2) for Microsoft Windows (32-bit)

B14270-01

Primary Author: Amitai Sela

Contributing Author: Laurel Hale, Cynthia Kibbe, Kishan Peyetti, Juan Ahues-Vasquez, Govind Lakkoju

Contributor: Orit Curiel, Jacco Draaijer, Vira Goorah

Contents

3 Microsoft SQL Server Gateway Features and Restrictions

4 Case Studies

Index

List of Tables

Send Us Your Comments

Oracle Transparent Gateway for Microsoft SQL Server Administrator's Guide, 10*g* Release 2 (10.2) for Microsoft Windows (32-bit)

B14270-01

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

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

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

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227. Attn: Server Technologies Documentation Manager
- Postal service:

  Oracle Corporation
  Server Technologies Documentation Manager
  500 Oracle Parkway, Mailstop 4op11
  Redwood Shores, CA 94065
  USA

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

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

Preface

This manual describes the Oracle Transparent Gateway for Microsoft SQL Server, which enables Oracle client applications to access Microsoft SQL Server 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 covers the following topics:

Audience

This manual is intended for Oracle database administrators who perform the following tasks:

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

> **Note:** You should understand the fundamentals of transparent gateways and the Microsoft Windows operating system before using this guide to install or administer the gateway.

Documentation Accessibility

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

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Organization

This document contains:

Chapter 1, "Introduction"

This chapter introduces the Oracle Transparent Gateway for Microsoft SQL Server.

Chapter 2, "Configuring the Gateway"

This chapter describes how to configure the gateway for Microsoft SQL Server.

Chapter 3, "Microsoft SQL Server Gateway Features and Restrictions"

This chapter describes how to use the gateway to access Microsoft SQL Server data, pass Microsoft SQL Server commands from applications to the Microsoft SQL Server database, perform distributed queries, and copy data.

Chapter 4, "Case Studies"

This chapter contains case studies that demonstrate some of the features of the Oracle Transparent Gateway.

Appendix A, "Data Type Conversion"

This appendix describes how the gateway converts Microsoft SQL Server data types to Oracle data types.

Appendix B, "Supported SQL Syntax and Functions"

This appendix describes the SQL statements and Oracle functions supported by Microsoft SQL Server.

Appendix C, "Data Dictionary"

This appendix contains information about data dictionary support, data dictionary mapping, and gateway data dictionary descriptions.

Appendix D, "Heterogeneous Services Initialization Parameters"

This appendix contains information about Heterogeneous Services initialization parameters.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Database New Features*
- *Oracle Call Interface Programmer's Guide*

- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database Application Developer's Guide - Fundamentals*
- *Oracle Database Concepts*
- *Oracle Database Performance Tuning Guide*
- *Oracle Database Error Messages*
- *Oracle Database Globalization Support Guide*
- *Oracle Database Reference*
- *Oracle Database SQL Reference*
- *Oracle Net Services Administrator's Guide*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database Heterogeneous Connectivity Administrator's Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Database Security Guide*

Many of the examples in this book use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://otn.oracle.com/technology/membership`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://otn.oracle.com/technology/documentation/`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
`UPPERCASE monospace (fixed-width) font`	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a `NUMBER` column. You can back up the database by using the `BACKUP` command. Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view. Use the `DBMS_STATS.GENERATE_STATS` procedure.
`lowercase monospace (fixed-width) font`	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to open SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true`. Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase italic monospace (fixed-width) font`	Lowercase italic monospace font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
`[ ]`	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`* `[ , `*`precision`* `])`
`{ }`	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`
`|`	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}` `[COMPRESS | NOCOMPRESS]`

Convention	Meaning	Example

...	Horizontal ellipsis points indicate either:
	■ That we have omitted parts of the code that are not directly related to the example
	■ That you can repeat a portion of the code
		`CREATE TABLE ... AS subquery;`
		
		`SELECT col1, col2, ... , coln FROM employees;`

.
.
.
	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	`SQL> SELECT NAME FROM V$DATAFILE;`
		`NAME`
		`----------`
		`/fsl/dbs/tbs_01.dbf`
		`/fs1/dbs/tbs_02.dbf`
		`.`
		`.`
		`.`
		`/fsl/dbs/tbs_09.dbf`
		`9 rows selected.`

Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	`acctbal NUMBER(11,2);` `acct CONSTANT NUMBER(4) := 3;`
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/system_password` `DB_NAME = database_name`

UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr/hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

Convention	Meaning	Example
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt\system32 is the same as C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"` `C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)`
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. For Microsoft Windows, the default location was `C:\orant`. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*nn*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle Database Platform Guide for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

1 Introduction

This chapter introduces the challenge faced by organizations when running several different database systems. It briefly covers Heterogeneous Services, the technology that the Oracle Transparent Gateway for Microsoft SQL Server is based on.

To get a good understanding of generic gateway technology, Heterogeneous Services, Generic Connectivity, and how Oracle Transparent gateways fit in the picture, reading *Oracle Database Heterogeneous Connectivity Administrator's Guide* first is highly recommended.

This chapter contains the following sections:

Overview

Heterogeneous data access is a problem that affects a lot of companies. A lot of companies run several different database systems. Each of these systems stores data and has a set of applications that run against it. Consolidation of this data in one database system is often hard-in large part because many of the applications that run against one database may not have an equivalent that runs against another. Until such time as migration to one consolidated database system is made feasible, it is necessary for the various heterogeneous database systems to interoperate.

Oracle Transparent Gateways provide the ability to transparently access data residing in a non-Oracle system from an Oracle environment. This transparency eliminates the need for application developers to customize their applications to access data from different non-Oracle systems, thus decreasing development efforts and increasing the mobility of the application. Applications can be developed using a consistent Oracle interface for both Oracle and Microsoft SQL Server.

Gateway technology is composed of two parts: a component that has the generic technology to connect to a non-Oracle system, which is common to all the non-Oracle systems, called Heterogeneous Services, and a component that is specific to the non-Oracle system that the gateway connects to. Heterogeneous Services, in conjunction with the Transparent Gateway agent, enables transparent access to non-Oracle systems from an Oracle environment.

Heterogeneous Services Technology

Heterogeneous Services provides the generic technology for connecting to non-Oracle systems. As an integrated component of the database, Heterogeneous Services can exploit features of the database, such as the powerful SQL parsing and distributed optimization capabilities.

Heterogeneous Services 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. Heterogeneous Services provides two types of translations: the ability to translate Oracle SQL into the proper dialect of the non-Oracle system as well as data dictionary translations which displays the metadata

of the non-Oracle system in the local format. For situations where no translations are available, native SQL can be issued to the non-Oracle system using the pass-through feature of Heterogeneous Services.

Heterogeneous Services also maintains the transaction coordination between Oracle and the remote non-Oracle system, such as providing the two-phase commit protocol to ensure distributed transaction integrity, even for non-Oracle systems that do not natively support two-phase commit.

> **See Also:** *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about Heterogeneous Services.

Oracle Transparent Gateways

The capabilities, SQL mappings, data type conversions, and interface to the remote non-Oracle system are contained in the gateway. The gateway interacts with Heterogeneous Services to provide the transparent connectivity between Oracle and non-Oracle systems.

The gateway doesn't have to be installed on the machine running Microsoft SQL Server. The gateway doesn't use SQL Server client, but uses MS-ODBC driver for SQL server that comes with all windows installations.

2 Configuring the Gateway

After installing the gateway, perform the following tasks to configure the gateway for Microsoft SQL Server:

Configuring the Gateway

Perform the following tasks to configure the Oracle Transparent Gateway for Microsoft SQL Server.

Task 1: Choose a System Identifier 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 Microsoft SQL Server database you are accessing. The SID is used as part of the file name for the initialization parameter file. The default SID is tg4msql.

You can define a gateway SID, but using the default of tg4msql is easier because you do not need to change the initialization parameter file name. However, if you want to access two Microsoft SQL Server databases, you need two gateway SIDs, one for each instance of the gateway. If you have one Microsoft SQL Server 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 Microsoft SQL Server 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\tg4msql\admin\inittg4msql.ora
```

Where ORACLE_HOME is the directory under which the gateway is installed.

If you are not using tg4msql 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 Microsoft SQL Server connection as follows:

```
HS_FDS_CONNECT_INFO=server_name.database_name
```

If you specify only *.database_name*, omitting *server_name* (but including the period), the gateway uses the following subtree of the Microsoft Windows registry to determine *server_name*:

HKEY_LOCAL_MACHINE\

SOFTWARE\

Microsoft\

MSSQLServer\

Client\

ConnectTo

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 D, "Heterogeneous Services Initialization Parameters" and *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about customizing the initialization parameter file.

Configuring Oracle Net Services Listener for the Gateway

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

Task 1: Configure Oracle Net Services TNS Listener for the Gateway

Oracle Net Services uses the TNS listener to receive incoming connections from a Oracle Net Services 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 directory under which the gateway is installed.

> **Note:** If Oracle Net Services 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 Oracle Net Services 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 Oracle Net Services 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:

Variable	Description
host_name	specifies 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=tg4msql)
      )
   )
```

Where:

Variable	Description
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.
`tg4msql`	specifies the Oracle Transparent Gateway for Microsoft SQL Server.

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=tg4msql)
   )
)
```

> **See Also:** *Oracle Net Services 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. Select Services from the Windows Control Panel. (For example, for Windows 2000, from the Windows Start menu select Settings, Control Panel, Administrative Tools, and then select Services).
2. Select the TNSListener service for the gateway.
3. If the service is already running, click Stop to stop it.
4. Click Start to start or restart the service.

Configuring the Oracle Database Server for Gateway Access

Before you use the gateway to access Microsoft SQL Server data you must configure the Oracle database server to enable communication with the gateway over Oracle Net Services.

Configuring Oracle Net Services for the Oracle Database Server

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

> **See Also:** *Oracle Database 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 Oracle Net Services 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:

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as tg4msql. 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 Microsoft Windows, 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 Oracle Net Services 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 2-2.
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 Oracle Net Services TNS Listener for the Gateway" on page 2-2 for more information.
`(HS=OK)`	specifies that this connect descriptor uses the Oracle Heterogeneous Services option.

Creating Database Links

Any Oracle client connected to the Oracle database server can access Microsoft SQL Server 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 Microsoft SQL Server 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 Microsoft SQL Server database.

> **See Also:** *Oracle Database Administrator's Guide* and *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about using database links.

Gateway Password Encryption Tool

The gateway uses userids and passwords to access the information in the remote database. Some userids and passwords must be defined in the Gateway Initialization

File to handle functions such as resource recovery. In the current security conscious environment, having plain-text passwords that are accessible in the Initialization File is deemed insecure. The tg4pwd encryption utility has been added as part of Heterogeneous Services' generic connectivity to help make this more secure. This utility is accessible by this gateway. The initialization parameters which contain sensitive values can be stored in an encrypted form.

> **See Also:** *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about using this utility.

Configuring the Gateway for Multiple Microsoft SQL Server Databases

The tasks for configuring the gateway to access multiple Microsoft SQL Server 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 `tg4msql`
- The gateway is configured for one Microsoft SQL Server database named `db1`
- Two Microsoft SQL Server databases named db2 and db3 on servers named `msql_nt2` and `msql_nt3` respectively are being added

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

- Configuring the gateway
- Configuring Oracle Net Services for the gateway and the Oracle database server

Multiple Databases Example: Configuring the Gateway

Choose Two System IDs for Each Microsoft SQL Server Database

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

- `tg4msql2` for the gateway accessing database db2
- `tg4msql3` 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\tg4msql\admin\inittg4msql.ora`, twice, naming one with the gateway SID for db2 and the other with the gateway SID for db3:

```
> cd ORACLE_HOME\tg4msql\admin
> copy inittg4msql.ora inittg4msql2.ora
> copy inittg4msql.ora inittg4msql3.ora
```

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

For `inittg4msql2.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=
msql_nt2
.db2
```

For `inittg4msql3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=
msql_nt3
.db3
```

> **Note:** If you have multiple gateway SIDs for the same Microsoft SQL Server 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 Oracle Net Services 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=tg4msql)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
   (SID_DESC=
      (SID_NAME=tg4msql2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
   (SID_DESC=
      (SID_NAME=tg4msql3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
)
```

Multiple Databases Example: Stopping and Starting the TNS Listener

1. Select Services from the Windows Control Panel. (For example, for Windows 2000, from the Windows Start menu select **Settings**,**Control Panel**, **Administrative Tools**, and then select **Services**.)
2. Select the TNSListener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

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

Configuring Oracle Net Services 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 Microsoft SQL Server example shows the entry for the original installed gateway first, followed by the two entries for the new gateway instances:

```
old_db_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=1541)
                 (HOST=gtwhost))
                 (CONNECT_DATA=
                     (SID=tg4msql))
                  (HS=OK))
new_db2_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=1541)
                 (HOST=gtwhost))
                 (CONNECT_DATA=
                     (SID=tg4msql2))
                 (HS=OK))
new_db3_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=1541)
                 (HOST=gtwhost))
                 (CONNECT_DATA=
                     (SID=tg4msql3))
                 (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 Microsoft SQL Server Data

Enter the following to create a database link for the `tg4msql2` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK MSQL2 CONNECT TO
  2  user2 IDENTIFIED BY password2 USING 'new_db2_using';
```

Enter the following to create a database link for the `tg4msql3` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK MSQL3 CONNECT TO
  2  user3 IDENTIFIED BY password3 USING 'new_db3_using';
```

Note: To encrypt the initialization parameters that would normally be stored in the initialization file in plain text, you must use the tg4pwd utility, as described in *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

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

```
SQL> SELECT * FROM ALL_USERS@MSQL2;
```

or

```
SQL> SELECT * FROM ALL_USERS@MSQL3;
```

Performing 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
- TWO_PHASE_COMMIT

By default, the gateway runs in COMMIT_CONFIRM transaction mode. When the Microsoft SQL Server database is updated by a transaction, the gateway becomes the commit point site. The Oracle database server commits the unit of work in the Microsoft SQL Server 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:** *Oracle Database Heterogeneous Connectivity Administrator's Guide* 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 by default **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 Microsoft SQL Server 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 do not use the default value RECOVER for the user name and password. Moreover, storing plain-text as user name and password in the initialization file is not a good security policy. There is now a utility called tg4pwd, that should be used for encryption. Refer to Chapter 4, 'Encrypting Initialization parameters' in the Heterogeneous Connectivity Administration Guide for further details.

1. Set up a user account in the Microsoft SQL Server database. Both the user name and password must be a valid Microsoft SQL Server 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 Microsoft SQL Server user account you set up for recovery.
   - HS_FDS_RECOVERY_PWD to the password of the Microsoft SQL Server user account you set up for recovery.

See Also: "Task 2: Customize the Initialization Parameter File" on page 2-1 for information about editing the initialization parameter file. For information about HS_FDS_RECOVERY_ACCOUNT and HS_FDS_RECOVERY_PWD, see Appendix D, "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 Microsoft SQL Server database for logging transactions. The gateway uses the transaction log table to check the status of failed transactions that were started at the Microsoft SQL Server database by the gateway and registered in the table.

Note: 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, called HS_TRANSACTION_LOG, consists of two columns, GLOBAL_TRAN_ID, data type CHAR(64) NOT NULL and TRAN_COMMENT, data type CHAR(255).

You can use another name for the log table, other than HS_TRANSACTION_LOG, by specifying the other name using the HS_FDS_TRANSACTION_LOG initialization parameter.

> **See Also:** Appendix D, "Heterogeneous Services Initialization Parameters" for information about the HS_FDS_TRANSACTION_LOG initialization parameter.

Create the transaction log table in the user account you created in "Task 1: Create a Recovery Account and Password" on page 2-9. Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database where the Microsoft SQL Server 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 `tg4msql_tx.sql` script, located in the directory `ORACLE_HOME\tg4msql\admin` where ORACLE_HOME is the directory under which the gateway is installed. Use isql to execute the script at the MS-DOS prompt, as follows:

```
> isql -Urecovery_account -Precovery_account_password [-Sserver] -itg4msql_tx.sql
```

Specifying an Owner

Instead of using the default owner name for the Microsoft SQL Server tables as defined in Microsoft SQL Server, 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 D, "Heterogeneous Services Initialization Parameters" for information about the HS_FDS_DEFAULT_OWNER initialization parameter.

3 Microsoft SQL Server Gateway Features and Restrictions

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

This chapter contains the following sections:

Using the Pass-Through Feature

The gateway can pass Microsoft SQL Server commands or statements from the application to the Microsoft SQL Server 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 Microsoft SQL Server database, as follows:

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

Where *command* cannot be one of the following:

- BEGIN TRANSACTION
- COMMIT
- ROLLBACK
- SAVE
- SHUTDOWN
- Microsoft SQL Server tool commands

The DBMS_HS_PASSTHROUGH package supports passing bind values and executing SELECT statements.

Note: TRUNCATE cannot be used in a pass-through statement.

See Also: *PL/SQL Packages and Types Reference* and Chapter 3 of Oracle Database Heterogeneous Connectivity Administrator's Guide for more information about the DBMS_HS_PASSTHROUGH package.

Executing Stored Procedures and Functions

Using the procedural feature, the gateway can execute stored procedures that are defined in the Microsoft SQL Server 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 user that the gateway is logging in as.

See Also: *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about executing stored procedures.

Standard PL/SQL statements are used to execute a stored procedure.

Remote User-defined Function Support

User-defined functions in a remote non-Oracle database can be used in SQL statements.

See Also: *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about executing user-defined functions on a Non-Oracle Database.

Return Values and Stored Procedures

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: Appendix D, "Heterogeneous Services Initialization Parameters" for information about both editing the initialization parameter file and the HS_FDS_PROC_IS_FUNC parameter.

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 Microsoft SQL Server stored procedure REVISE_SALARY. The stored procedure retrieves the salary value from the Microsoft SQL Server 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';
  RESULT := REVISE_SALARY@
```

```
MSQL
(INPUT);
  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.

Result Sets and Stored Procedures

The Oracle Transparent Gateway for Microsoft SQL Server provides support for stored procedures which return result sets.

By default, all stored procedures and functions do not return a result set to the user. To enable result sets, set the HS_FDS_RESULTSET_SUPPORT parameter in the initialization parameter file.

> **See Also:** Appendix D, "Heterogeneous Services Initialization Parameters" for information about both editing the initialization parameter file and the HS_FDS_RESULTSET_SUPPORT parameter. For further information about Oracle support for result sets in non-Oracle databases see *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

Note: If you set the HS_FDS_RESULTSET_SUPPORT gateway initialization parameter, you must change the syntax of the procedure execute statement for all existing stored procedures or errors will occur.

When accessing stored procedures with result sets through the Oracle Transparent Gateway for Microsoft SQL Server, you must work in the sequential mode of Heterogeneous Services.

The Oracle Transparent Gateway for Microsoft SQL Server returns the following information to Heterogeneous Services during procedure description:

- All the input arguments of the remote stored procedure
- None of the output arguments
- One out argument of type ref cursor (corresponding to the first result set returned by the stored procedure)

Client programs have to use the virtual package function dbms_hs_result_set.get_next_result_set to get the ref cursor for subsequent result sets. The last result set returned is the out argument from the procedure.

The limitations of accessing result sets are the following:

- Result sets returned by a remote stored procedure have to be retrieved in the order in which they were placed on the wire
- When executing multiple stored procedures, the result set of each procedure must be retrieved in full before executing the next stored procedure

- On execution of a stored procedure, all result sets returned by a previously executed stored procedure will be closed (regardless of whether the data has been completely retrieved or not)

In the following example, the Microsoft SQL Server stored procedure is executed to fetch the contents of the emp and dept tables from Microsoft SQL Server:

```
create procedure REFCURPROC (@arg1 varchar(255), @arg2 varchar(255) output)
as
select @arg2 = @arg1
select * from EMP
select * from DEPT
go
```

This stored procedure assigns the input parameter arg1 to the output parameter arg2, opens the query `SELECT * FROM EMP` in ref cursor rc1, and opens the query `SELECT * FROM DEPT` in ref cursor rc2.

OCI Program Fetching from Result Sets in Sequential Mode

The following example shows OCI program fetching from result sets in sequential mode:

```
OCIEnv *ENVH;
OCISvcCtx *SVCH;
OCIStmt *STMH;
OCIError *ERRH;
OCIBind *BNDH[3];
OraText arg1[20];
OraText arg2[255];
OCIResult *rset;
OCIStmt *rstmt;
ub2 rcode[3];
ub2 rlens[3];
sb2 inds[3];
OraText *stmt = (OraText *) "begin refcurproc@MSQL(:1,:2,:3); end;";
OraText *n_rs_stm = (OraText *)
  "begin :ret := DBMS_HS_RESULT_SET.GET_NEXT_RESULT_SET@MSQL; end;";

/* Prepare procedure call statement */

/* Handle Initialization code skipped */
OCIStmtPrepare(STMH, ERRH, stmt, strlen(stmt), OCI_NTV_SYNTAX, OCI_DEFAULT);

/* Bind procedure arguments */
inds[0] = 0;
strcpy((char *) arg1, "Hello World");
rlens[0] = strlen(arg1);
OCIBindByPos(STMH, &BNDH[0], ERRH, 1, (dvoid *) arg1, 20, SQLT_CHR,
             (dvoid *) &(inds[0]), &(rlens[0]), &(rcode[0]), 0, (ub4 *) 0,
             OCI_DEFAULT);

inds[1] = -1;
OCIBindByPos(STMH, &BNDH[1], ERRH, 1, (dvoid *) arg2, 20, SQLT_CHR,
             (dvoid *) &(inds[1]), &(rlens[1]), &(rcode[1]), 0, (ub4 *) 0,
             OCI_DEFAULT);

inds[2] = 0;
rlens[2] = 0;
OCIDescriptorAlloc(ENVH, (dvoid **) &rset, OCI_DTYPE_RSET, 0, (dvoid **) 0);
OCIBindByPos(STMH, &BNDH[2], ERRH, 2, (dvoid *) rset, 0, SQLT_RSET,
```

```
             (dvoid *) &(inds[2]), &(rlens[2]), &(rcode[2]),
             0, (ub4 *) 0, OCI_DEFAULT);

/* Execute procedure */
OCIStmtExecute(SVCH, STMH, ERRH, 1, 0, (CONST OCISnapshot *) 0,
               (OCISnapshot *) 0, OCI_DEFAULT);

/* Convert result set to statement handle */
OCIResultSetToStmt(rset, ERRH);
rstmt = (OCIStmt *) rset;

/* After this the user can fetch from rstmt */
/* Issue get_next_result_set call to get handle to next_result set */
/* Prepare Get next result set procedure call */

OCIStmtPrepare(STMH, ERRH, n_rs_stm, strlen(n_rs_stm), OCI_NTV_SYNTAX,
               OCI_DEFAULT);

/* Bind return value */
OCIBindByPos(STMH, &BNDH[1], ERRH, 1, (dvoid *) rset, 0, SQLT_RSET,
             (dvoid *) &(inds[1]), &(rlens[1]), &(rcode[1]),
             0, (ub4 *) 0, OCI_DEFAULT);

/* Execute statement to get next result set*/
OCIStmtExecute(SVCH, STMH, ERRH, 1, 0, (CONST OCISnapshot *) 0,
               (OCISnapshot *) 0, OCI_DEFAULT);

/* Convert next result set to statement handle */
OCIResultSetToStmt(rset, ERRH);
rstmt = (OCIStmt *) rset;

/* Now rstmt will point to the second result set returned by the
remote stored procedure */

/* Repeat execution of get_next_result_set to get the output arguments */
```

PL/SQL Program Fetching from Result Sets in Sequential Mode

Assume that the table loc_emp is a local table exactly like the Microsoft SQL Server emp table. The same assumption applies for loc_dept. outargs is a table with columns corresponding to the out arguments of the Microsoft SQL Server stored procedure.

```
declare
  rc1 rcpackage.rctype;
  rec1 loc_emp%rowtype;
  rc2 rcpackage.rctype;
  rec2 loc_dept%rowtype;
  rc3 rcpackage.rctype;
  rec3 outargs%rowtype;
  out_arg varchar2(255);

begin

  -- Execute procedure
  out_arg := null;
  refcurproc@MSQL('Hello World', out_arg, rc1);

  -- Fetch 20 rows from the remote emp table and insert them into loc_emp
  for i in 1 .. 20 loop
    fetch rc1 into rec1;
    insert into loc_emp (rec1.empno, rec1.ename, rec1.job,
```

```
    rec1.mgr, rec1.hiredate, rec1.sal, rec1.comm, rec1.deptno);
  end loop;

  -- Close ref cursor
  close rc1;

  -- Get the next result set returned by the stored procedure
  rc2 := dbms_hs_result_set.get_next_result_set@MSQL;

  -- Fetch 5 rows from the remote dept table and insert them into loc_dept
  for i in 1 .. 5 loop
    fetch rc2 into rec2;
    insert into loc_dept values (rec2.deptno, rec2.dname, rec2.loc);
  end loop;

  --Close ref cursor
  close rc2;

  -- Get the output arguments from the remote stored procedure
  -- Since we are in sequential mode, they will be returned in the
  -- form of a result set
  rc3 := dbms_hs_result_set.get_next_result_set@MSQL;

  -- Fetch them and insert them into the outarguments table
  fetch rc3 into rec3;
  insert into outargs (rec3.outarg, rec3.retval);

  -- Close ref cursor
  close rc3;

end;
/
```

Database Compatibility Issues for Microsoft SQL Server

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

- Implicit Transactions (Chained Mode)
- Column Definitions
- Naming Rules
- Data Types
- Queries
- Locking

Implicit Transactions (Chained Mode)

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

Column Definitions

By default, a Microsoft SQL Server table column cannot contain null values unless NULL is specified in the column definition. Microsoft SQL Server assumes all columns cannot contain null values unless you set a Microsoft SQL Server option to override this default.

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 Microsoft SQL Server 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 quotation marks, case sensitivity, and the use of alphanumeric characters can all be different.

> **See Also:** *Oracle Database Reference* and Microsoft SQL Server documentation.

Case Sensitivity

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

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

However, to refer to the Microsoft SQL Server table called emp owned by Scott from an Oracle application, enter the following:

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

If the Microsoft SQL Server 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"@MSQL;
```

Or

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

Oracle recommends that you surround all Microsoft SQL Server object names with double quote characters and use the exact letter case for the object names as they appear in the Microsoft SQL Server data dictionary. This convention is not required when referring to the supported Oracle data dictionary tables or views listed in Appendix C, "Data Dictionary".

If existing applications cannot be changed according to these conventions, create views in Oracle to associate Microsoft SQL Server names to the correct letter case. For example, to refer to the Microsoft SQL Server 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"@MSQL;
```

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 Microsoft SQL Server data dictionary. You must be prepared to update the Oracle view definitions whenever the data definitions for the corresponding tables are changed in the Microsoft SQL Server database.

Data Types

Data type issues include the following:

- Binary Literal Notation
- Bind Variables With LONG Columns
- 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 Microsoft SQL Server 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@MSQL 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.

Bind Variables With LONG Columns

The gateway does not support using bind variables to update columns of data type LONG.

Data Type Conversion

Microsoft SQL Server 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@
MSQL
 WHERE DATE_COL = "1-JAN-2004";
```

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

```
SELECT DATE_COL FROM TEST@
MSQL
 WHERE DATE_COL = TO_DATE("1-JAN-2004")
```

See Also: Appendix A, "Data Type Conversion" for more information about restrictions on data types.

Queries

Query issues include the following:

- Row Selection
- Empty Strings
- Empty Bind Variables

Row Selection

Microsoft SQL Server 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. Microsoft SQL Server processes an empty string as an empty string.

Comparing to an empty string

The gateway passes the empty string to the Microsoft SQL Server database without any conversion. If you intended an empty string to represent a null value, Microsoft SQL Server does not process the statement that way; it uses the empty string.

You can 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 * from "emp"@MSQL where "ename" IS NULL;
```

Selecting an empty string

For VARCHAR columns, the gateway returns an empty string to the Oracle database server as NULL value.

For CHAR columns, the gateway returns the full size of the column with each character as empty space (' ').

Empty Bind Variables

For VARCHAR bind variables, the gateway passes empty bind variables to the Microsoft SQL Server database as a NULL value.

Locking

The locking model for an Microsoft SQL Server database differs significantly from the Oracle model. The gateway depends on the underlying Microsoft SQL Server behavior, so Oracle applications that access Microsoft SQL Server 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: Microsoft SQL Server documentation for information about the Microsoft SQL Server locking model.

Known Restrictions

If you encounter incompatibility problems not listed in this section or in "Known Problems" on page 3-14, please contact Oracle Support Services. The following section describes the known restrictions and includes suggestions for dealing with them when possible:

- Multiple Open Statements
- Transactional Integrity
- Transaction Capability
- COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors
- Stored Procedures
- Pass-Through Feature
- DDL Statements
- SQL Syntax
- Functions
- SQL*Plus COPY Command with Lowercase Table Names
- Database Links

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

Multiple Open Statements

The gateway uses Microsoft ODBC driver to talk to Microsoft SQL Server. Microsoft's ODBC driver has the limitation that only one open statement or cursor is allowed for each connection. If a second statement or cursor needs to open in the same transaction to access SQL Server, it requires a new connection.

Because of this limitation multiple open statements or cursors within the same transaction can lock each other because they use different connections to Microsoft SQL Server.

To avoid this restriction, issue a commit, or modify the logic, or both.

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 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.

The gateway sets Autocommit Mode to Off when a connection is established to the Microsoft SQL Server database.

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.

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.

When accessing stored procedures with result sets through the Oracle Transparent Gateway for Microsoft SQL Server, you must work in the sequential mode of Heterogeneous Services.

When accessing stored procedures with multiple result sets through the Oracle Transparent Gateway for Microsoft SQL Server, you must read all the result sets before continuing.

Output parameters of stored procedures must be initialized to an empty string.

Pass-Through Feature

If the SQL statements being passed through the gateway result in an implicit commit at the Microsoft SQL Server database, the Oracle transaction manager is unaware of the commit and an Oracle ROLLBACK command cannot be used to roll back the transaction.

DDL Statements

Microsoft SQL Server requires some DDL statements to be executed in their own transaction, and only one DDL statement can be executed in a given transaction.

If you use these DDL statements in a Microsoft SQL Server stored procedure and you execute the stored procedure through the gateway using the procedural feature, or, if you execute the DDL statements through the gateway using the pass-through feature, an error condition might result. This is because the procedural feature and the pass-through feature of the gateway cannot guarantee that the DDL statements are executed in their own separate transaction.

The following Microsoft SQL Server DDL statements can cause an error condition if you attempt to pass them with the gateway pass-through feature, or if you execute a Microsoft SQL Server stored procedure that contains them:

Table 3–1 Restricted DDL Statements

Statement Name
ALTER DATABASE
CREATE DATABASE
CREATE INDEX
CREATE PROCEDURE
CREATE TABLE
CREATE VIEW
DISK INIT
DROP <*object*>
DUMP TRANSACTION
GRANT
LOAD DATABASE
LOAD TRANSACTION
RECONFIGURE
REVOKE
SELECT INTO
TRUNCATE TABLE
UPDATE STATISTICS

See Also: Microsoft SQL Server documentation for more information about DDL statements.

SQL Syntax

This section lists restrictions on the following SQL syntax:

- WHERE CURRENT OF Clause
- CONNECT BY Clause
- Functions in Subqueries
- Parameters in Subqueries
- Data Dictionary Table and Views in UPDATE Statement
- ROWID

- TO_DATE
- EXPLAIN PLAN Statement

> **See Also:** Appendix B, "Supported SQL Syntax and Functions" for more information about restrictions on SQL syntax.

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.

Functions in Subqueries

Bind variables and expressions are not supported as operands in string functions or mathematical functions, when part of subquery in an INSERT, UPDATE or DELETE SQL statement.

Parameters in Subqueries

Due to a limitation in Microsoft SQL Server, you cannot use parameters in subqueries.

Data Dictionary Table and Views in UPDATE Statement

Data dictionary tables and views in the SET clause of an UPDATE statement are not supported.

ROWID

The Oracle ROWID implementation is not supported.

TO_DATE

TO_DATE is a reserved word and cannot be used as a database identifier name.

EXPLAIN PLAN Statement

The EXPLAIN PLAN statement is not supported.

Functions

The following restrictions apply to using functions:

- Unsupported functions cannot be used in statements that refer to LONG columns.
- 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 Microsoft SQL Server.

SQL*Plus COPY Command with Lowercase Table Names

You need to use double quotes to wrap around lower case table names.

For example:

```
copy from tkhouser/tkhouser@inst1 insert loc_tkhodept using select * from
```

```
"tkhodept"@holink2;
```

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.

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 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
- Date Arithmetic
- Microsoft SQL Server IMAGE, TEXT and NTEXT 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 Oracle9*i* database server (Release 9.2 and earlier) supported 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 Oracle9*i* database server, you must change the setting to FALSE, the default setting, to allow Oracle9*i* to communicate with the gateway.

In the current release, Oracle Database 10*g*, Release 10.2, the DBLINK_ENCRYPT_LOGIN initialization parameter is obsolete, so you need not check it.

Date Arithmetic

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

```
date + number
number + date
```

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

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

Microsoft SQL Server IMAGE, TEXT and NTEXT Data Types

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

- An unsupported SQL function cannot be used in a SQL statement that accesses a column defined as Microsoft SQL Server data type IMAGE, TEXT or NTEXT.
- You cannot use SQL*Plus to select data from a column defined as Microsoft SQL Server data type IMAGE, TEXT or NTEXT when the data is greater than 80 characters in length. Oracle recommends using Pro*C or Oracle Call Interface to access such data in a Microsoft SQL Server database.
- IMAGE, TEXT and NTEXT data types must be NULLABLE for INSERT or UPDATE to work.
- A table including an IMAGE, TEXT or NTEXT 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, TEXT and NTEXT data cannot be read through pass-through queries.
- If a SQL statement is accessing a table including an IMAGE, TEXT or NTEXT column, the statement will be sent to Microsoft SQL Server as two separate statements. One statement to access the IMAGE, TEXT or NTEXT column, and a second statement for the other columns in the original statement. This will result in two connections to Microsoft SQL Server due to a limitation in the Microsoft ODBC driver which only allows one statement for each connection, which can cause a hang depending on the sequence of SQL statements. If this happens, try issuing a commit and separating the statements in different transactions.

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

> **See Also:** Appendix B, "Supported SQL Syntax and Functions" for more information about restrictions on SQL syntax.

String Functions

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

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

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

Schema Names and PL/SQL

If you do not prefix a Microsoft SQL Server 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

When executing multiple stored procedures, the result set of each procedure must be retrieved in full before executing the next stored procedure.

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.

4 Case Studies

The following case studies for Microsoft SQL Server 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

The cases illustrate:

- A simple query (Case 1)
- A more complex query (Case 2)
- Joining Microsoft SQL Server tables (Case 3)
- Write capabilities (Case 4)
- A data dictionary query (Case 5)
- The pass-through feature (Case 6)
- Executing stored procedures (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 Microsoft SQL Server database

- One SQL script file that drops the demonstration tables and stored procedures from the Microsoft SQL Server database

Demonstration Files

After a successful gateway installation, use the demonstration files stored in the directory `ORACLE_HOME\tg4msql\demo` where ORACLE_HOME is the directory under which the gateway is installed. The directory contains the following demonstration files:

Demonstration Files
bldmsql.sql
case1.sql
case2.sql
case3.sql
case4a.sql
case4b.sql
case4c.sql
case5.sql
case6a.sql
case6b.sql
case7.sql
dropmsql.sql

Demonstration Requirements

The case studies assume these requirements have been met:

- The gateway demonstration tables and stored procedures are installed in the Microsoft SQL Server 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 Microsoft SQL Server 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';
```

- Oracle Net Services is configured correctly and running

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 Microsoft SQL Server database, use the `bldmsql.sql` script to create them. Enter the following:

```
> isql -USCOTT -PTIGER2 -ibldmsql.sql
```

The script creates the demonstration tables and stored procedures in the Microsoft SQL Server 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 following table definitions use information retrieved by the SQL*PLUS 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 Microsoft SQL Server 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.

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.

Case 3: Joining Microsoft SQL Server Tables

Case 3 demonstrates the following:

- Joins between Microsoft SQL Server tables
- Subselects

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

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.

UPDATE Statement

Case 4b provides an example of a simple UPDATE statement. In this example, employees are given a $100 a month salary increase.

INSERT Statement

Case 4c is an example of a simple insert statement that does not provide information for all columns.

Case 5: Data Dictionary Query

Case 5 demonstrates data dictionary mapping. It retrieves all the tables and views that exist in the Microsoft SQL Server database that begin with "GTW".

Case 6: The Pass-Through Feature

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

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.

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.

Case 7: Executing Stored Procedures

Case 7 demonstrates the gateway executing a stored procedure in the Microsoft SQL Server database.

A Data Type Conversion

This appendix contains the following section:

Data Type Conversion

The gateway converts Microsoft SQL Server data types to Oracle data types as follows:

Table A–1 Data Type Conversions

Microsoft SQL Server	Oracle	Comment
BIGINT	NUMBER(19)	
BINARY	RAW	-
BIT	NUMBER(3)	-
CHAR	CHAR	-
DATETIME	DATE	Fractional parts of a second are truncated
DECIMAL	NUMBER(p[,s])	-
FLOAT	FLOAT(49)	-
IMAGE	LONG RAW	-
INTEGER	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
MONEY	NUMBER(19,4)	-
NCHAR	NCHAR	-
NTEXT	LONG	-
NVARCHAR	NCHAR	-
NUMERIC	NUMBER(p[,s])	-
REAL	FLOAT(23)	-
SMALL DATETIME	DATE	The value for seconds is returned as 0
SMALL MONEY	NUMBER(10,4)	-
SMALLINT	NUMBER(5)	NUMBER range is -32,767 to 32,767
TEXT	LONG	-
TIMESTAMP	RAW	-
TINYINT	NUMBER(3)	-
UNIQUEIDENTIFIER	CHAR(36)	
VARBINARY	RAW	-
VARCHAR	VARCHAR2	-

B Supported SQL Syntax and Functions

This appendix contains the following sections:

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 Microsoft SQL Server database.

Note: TRUNCATE cannot be used in a pass-through statement.

See Also: *Oracle Database SQL Reference* for a detailed descriptions of keywords, parameters, and options.

DELETE

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

See Also: "Functions Supported by Microsoft SQL Server" on page B-2 for a list of supported functions.

INSERT

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

See Also: "Functions Supported by Microsoft SQL Server" on page B-2 for a list of supported functions.

SELECT

The SELECT statement is fully supported, with these exceptions:

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

UPDATE

The UPDATE statement is fully supported. However, only Oracle functions supported by Microsoft SQL Server can be used.

> **See Also:** "Functions Supported by Microsoft SQL Server" on page B-2 for a list of supported functions.

Oracle Functions

All functions are evaluated by the Microsoft SQL Server database after the gateway has converted them to Microsoft SQL Server SQL equivalents. The exception is the TO_DATE function, which is evaluated by the gateway.

Functions Not Supported by Microsoft SQL Server

Oracle SQL functions with no equivalent function in Microsoft SQL Server 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 Microsoft SQL Server

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

- Arithmetic Operators
- Comparison Operators
- Pattern Matching
- Group Functions
- String Functions
- Other Functions

Arithmetic Operators

Oracle	Microsoft SQL Server
+	+
-	-
*	*
/	/

Comparison Operators

Oracle	Microsoft SQL Server
=	=
>	>

Oracle	Microsoft SQL Server
<	<
>=	>=
<=	<=
<>, !=, ^=	<>
IS NOT NULL	IS NOT NULL
IS NULL	IS NULL

Pattern Matching

Oracle	Microsoft SQL Server
LIKE	LIKE
NOT LIKE	NOT LIKE

Group Functions

Oracle	Microsoft SQL Server
AVG	AVG
COUNT	COUNT
MAX	MAX
MIN	MIN
SUM	SUM

String Functions

Oracle	Microsoft SQL Server
||, CONCAT	+ (*expression1* + *expression2*)
ASCII	ASCII
CHR	CHAR
INSTR (with two arguments)	CHARINDEX
LENGTH	DATALENGTH
LOWER	LOWER
LTRIM	LTRIM
RTRIM	RTRIM
SUBSTR (second argument cannot be a negative number)	SUBSTRING
UPPER	UPPER

Other Functions

Oracle	Microsoft SQL Server
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

Functions Supported by the Gateway

If an Oracle function has no equivalent function in Microsoft SQL Server, the Oracle function is not translated into the SQL statement and must be post-processed if the SQL statement is a SELECT.

The gateway, however, does support the TO_DATE function equivalent in Microsoft SQL Server, as follows:

`TO_DATE(`*`date_string`* | *`date_column`*`)`
Where:

date_string is converted to a string with the following format:

`yyyy-mm-dd hh:mi:ss.fff`

> **Recommendation:** Supply the date string with the same format as the result (that is, *`yyyyy-mm-dd hh:mi:ss.fff`*).

`date_column` is a column with a date data type. It is converted to a parameter with a timestamp data type.

C Data Dictionary

The Oracle Transparent Gateway for Microsoft SQL Server translates a query that refers to an Oracle database server data dictionary table into a query that retrieves the data from Microsoft SQL Server system tables. 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

The following paragraphs describe the Oracle Transparent Gateway for Microsoft SQL Server data dictionary support.

Microsoft SQL Server System Tables

Microsoft SQL Server data dictionary information is stored in the Microsoft SQL Server database as Microsoft SQL Server system tables. All Microsoft SQL Server system tables have names prefixed with "sys". The Microsoft SQL Server system tables define the structure of a database. When you change data definitions, Microsoft SQL Server reads and modifies the Microsoft SQL Server 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_CATALOG` to retrieve all table names in the Microsoft SQL Server database:

```
SQL> SELECT * FROM "ALL_CATALOG"@MSQL;
```

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

1. Maps the requested table, view, or synonym to one or more Microsoft SQL Server system table names. The gateway translates all data dictionary column names to their corresponding Microsoft SQL Server column names within the query. If the mapping involves one Microsoft SQL Server system table, the gateway translates the requested table name to its corresponding Microsoft SQL Server system table name within the query. If the mapping involves multiple Microsoft SQL Server system tables, the gateway constructs a join in the query using the translated Microsoft SQL Server system table names.
2. Sends the translated query to Microsoft SQL Server.

3. Might convert the retrieved Microsoft SQL Server data to give it the appearance of the Oracle database server data dictionary table.
4. Passes the data dictionary information from the translated Microsoft SQL Server 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 SQL*Plus DESCRIBE command.

Direct Queries to Microsoft SQL Server Tables

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

Supported Views and Tables

The gateway supports the following views and tables:

Supported Views and Table	Supported Views and Table
ALL_CATALOG	ALL_COL_COMMENTS
ALL_CONS_COLUMNS	ALL_CONSTRAINTS
ALL_IND_COLUMNS	ALL_INDEXES
ALL_OBJECTS	ALL_TAB_COLUMNS
ALL_TAB_COMMENTS	ALL_TABLES
ALL_USERS	ALL_VIEWS
DBA_CATALOG	DBA_COL_COMMENTS
DBA_OBJECTS	DBA_TAB_COLUMNS
DBA_TAB_COMMENTS	DBA_TABLES
DICT_COLUMNS	DICTIONARY
DUAL	TABLE_PRIVILEGES
USER_CATALOG	USER_COL_COMMENTS
USER_CONS_COLUMNS	USER_CONSTRAINTS
USER_IND_COLUMNS	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 any data dictionary table or view beginning with ALL_ can returns rows from the Microsoft SQL Server database even when access privileges for those Microsoft SQL Server objects have not been granted. 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 Microsoft SQL Server system tables used. A plus sign (+) indicates that a join operation is involved.

Table C–1 Oracle Data Dictionary View Names and Microsoft SQL Server Equivalents

View Name	Microsoft SQL Server System Table Name
ALL_CATALOG	sysusers + sysobjects
ALL_COL_COMMENTS	sysusers+sysobjects+syscolumns
ALL_CONS_COLUMNS	sp_pkeys + sp_fkeys
ALL_CONSTRAINTS	sysusers + sysobjects + sysindexes + sysconstraints + sysreferences
ALL_IND_COLUMNS	sysusers + sysindexes + syscolumns
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_PRIVILEGES	sysprotects + sysusers + sysobjects
USER_CATALOG	sysusers + sysobjects
USER_COL_COMMENTS	sysusers + sysobjects + syscolumns
USER_CONS_COLUMNS	sp_pkeys + sp_fkeys
USER_CONSTRAINTS	sysusers + sysobjects + sysindexes + sysconstraints + sysreferences
USER_IND_COLUMNS	sysusers + sysindexes + syscolumns
USER_INDEXES	sysusers + sysindexes + sysobjects
USER_OBJECTS	sysusers + sysobjects + sysindexes

Table C–1 (Cont.) Oracle Data Dictionary View Names and Microsoft SQL Server

View Name	Microsoft SQL Server System Table Name
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 an Microsoft SQL Server system 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, data type, and width of each column
- The contents of columns with fixed values

They are described here with information retrieved by an Oracle SQL*Plus 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.

> **Note:** The column width of some columns in the translated data dictionary tables would be different when the gateway connects to a Microsoft SQL Server Version 7.0 database.

Table C–2 ALL_CATALOG

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"

Table C–3 ALL_COL_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
COMMENTS	-	VARCHAR2(1)	-

Table C–4 ALL_CONS_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
CONSTRAINT_NAME	-	VARCHAR2(30)	-
TABLE_NAME	-	VARCHAR2(30)	-
COLUMN_NAME	-	VARCHAR2(8192)	-
POSITION	-	FLOAT(49)	-

Table C–5 ALL_CONSTRAINTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
CONSTRAINT_NAME	-	VARCHAR2(256)	-
CONSTRAINT_TYPE	-	VARCHAR2(1)	"C" or "P" or "R" or "U"
TABLE_NAME	-	VARCHAR2(256)	-
SEARCH_CONDITION	-	VARCHAR2(1)	NULL
R_OWNER	-	VARCHAR2(256)	-
R_CONSTRAINT_NAME	-	VARCHAR2(256)	-
DELETE_RULE	-	VARCHAR2(1)	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	-

Table C–6 ALL_IND_COLUMNS

Name	Null?	Type	Value
INDEX_OWNER	NOT NULL	VARCHAR2(30)	-
INDEX_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
COLUMN_NAME	-	VARCHAR2(8192)	-
COLUMN_POSITION	NOT NULL	FLOAT(49)	-

Table C–6 (Cont.) ALL_IND_COLUMNS

Name	Null?	Type	Value
COLUMN_LENGTH	NOT NULL	FLOAT(49)	-
DESCEND	-	VARCHAR2(4)	-

Table C–7 ALL_INDEXES

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
INDEX_NAME	-	VARCHAR2(256)	-
INDEX_TYPE	-	VARCHAR2(1)	NULL
TABLE_OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
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
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

Table C–7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
NUM_ROWS	-	NUMBER	0
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	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
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 C–8 ALL_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
OBJECT_NAME	-	VARCHAR2(256)	-
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 C–8 (Cont.) ALL_OBJECTS

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

Table C–9 ALL_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
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	-	DATE	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 C–10 ALL_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-

Table C–10 (Cont.) ALL_TAB_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS	-	VARCHAR2(1)	NULL

Table C–11 ALL_TABLES

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
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
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 C–11 (Cont.) ALL_TABLES

Name	Null?	Type	Value
TABLE_LOCK	-	VARCHAR2(1)	NULL
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	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 C–12 ALL_USERS

Name	Null?	Type	Value
USERNAME	-	VARCHAR2(256)	-
USER_ID	NOT NULL	NUMBER(5)	-
CREATED	NOT NULL	DATE	-

Table C–13 ALL_VIEWS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
VIEW_NAME	-	VARCHAR2(256)	-
TEXT_LENGTH	-	NUMBER	0
TEXT	-	VARCHAR2(256)	-
TYPE_TEXT_LENGTH	-	NUMBER	0
TYPE_TEXT	-	VARCHAR2(1)	-
OID_TEXT_LENGTH	-	NUMBER	0
OID_TEXT	-	VARCHAR2(1)	-
VIEW_TYPE_OWNER	-	VARCHAR2(1)	-
VIEW_TYPE	-	VARCHAR2(1)	-

Table C–14 DBA_CATALOG

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"

Table C–15 DBA_COL_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
COMMENTS	-	VARCHAR2(1)	NULL

Table C–16 DBA_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
OBJECT_NAME	-	VARCHAR2(256)	-
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 C–17 DBA_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-

Table C–17 (Cont.) DBA_TAB_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
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	-	DATE	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 C–18 DBA_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS	-	VARCHAR2(1)	NULL

Table C–19 DBA_TABLES

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
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
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	-	DATE	NULL
PARTITIONED	-	VARCHAR2(1)	NULL
IOT_TYPE	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARHCAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL
NESTED	-	VARCHAR2(1)	NULL

Table C–19 (Cont.) DBA_TABLES

Name	Null?	Type	Value
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 C–20 DICT_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
COMMENTS	-	VARCHAR2(1)	NULL

Table C–21 DICTIONARY

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
COMMENTS	-	VARCHAR2(1)	-

Table C–22 DUAL

Name	Null?	Type	Value
DUMMY	NOT NULL	VARCHAR2(1)	"X"

Table C–23 TABLE_PRIVILEGES

Name	Null?	Type	Value
GRANTEE	-	VARCHAR2(256)	-
OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
GRANTOR	-	VARCHAR2(256)	-
SELECT_PRIV	-	VARCHAR2(1)	"Y"
INSERT_PRIV	-	VARCHAR2(1)	"A"
DELETE_PRIV	-	VARCHAR2(1)	"Y"
UPDATE_PRIV	-	VARCHAR2(1)	"A"

Table C–23 (Cont.) TABLE_PRIVILEGES

Name	Null?	Type	Value
REFERENCES_PRIV	-	VARCHAR2(1)	"A"
ALTER_PRIV	-	VARCHAR2(1)	"Y"
INDEX_PRIV	-	VARCHAR2(1)	"Y"
CREATED	NOT NULL	DATE	-

Table C–24 USER_CATALOG

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"

Table C–25 USER_COL_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
COMMENTS	-	VARCHAR2(1)	NULL

Table C–26 USER_CONS_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
COLUMN_NAME	-	VARCHAR2(8192)	-
POSITION	-	FLOAT(49)	-

Table C–27 USER_CONSTRAINTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(256)	-
CONSTRAINT_NAME	-	VARCHAR2(256)	-
CONSTRAINT_TYPE	-	VARCHAR2(1)	"R" or "P" or "U" or "C"
TABLE_NAME	-	VARCHAR2(256)	-
SEARCH_CONDITION	-	VARCHAR2(1)	NULL

Table C–27 (Cont.) USER_CONSTRAINTS

Name	Null?	Type	Value
R_OWNER	-	VARCHAR2(256)	-
R_CONSTRAINT_NAME	-	VARCHAR2(256)	-
DELETE_RULE	-	VARCHAR2(1)	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	-

Table C–28 USER_IND_COLUMNS

Name	Null?	Type	Value
INDEX_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
COLUMN_NAME	-	VARCHAR2(8192)	-
COLUMN_POSITION	NOT NULL	FLOAT(49)	-
COLUMN_LENGTH	NOT NULL	FLOAT(49)	-
DESCEND	-	VARCHAR2(4)	-

Table C–29 USER_INDEXES

Name	Null?	Type	Value
INDEX_NAME	-	VARCHAR2(256)	-
INDEX_TYPE	-	VARCHAR2(1)	NULL
TABLE_OWNER	-	VARCHAR2(256)	-
TABLE_NAME	-	VARCHAR2(256)	-
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

Table C–29 (Cont.) USER_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	-	DATE	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
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

Table C–29 (Cont.) USER_INDEXES

Name	Null?	Type	Value
FUNCIDX_STATUS	-	VARCHAR2(1)	NULL

Table C–30 USER_OBJECTS

Name	Null?	Type	Value
OBJECT_NAME	-	VARCHAR2(256)	-
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 C–31 USER_TAB_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
COLUMN_NAME	-	VARCHAR2(256)	-
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

Table C–31 (Cont.) USER_TAB_COLUMNS

Name	Null?	Type	Value
HIGH_VALUE	-	NUMBER	0
DENSITY	-	NUMBER	0
NUM_NULLS	-	NUMBER	0
NUM_BUCKETS	-	NUMBER	0
LAST_ANALYZED	-	DATE	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 C–32 USER_TAB_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS	-	VARCHAR2(1)	NULL

Table C–33 USER_TABLES

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(256)	-
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

Table C–33 (Cont.) USER_TABLES

Name	Null?	Type	Value
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	-	DATE	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 C–34 USER_USERS

Name	Null?	Type	Value
USERNAME	-	VARCHAR2(256)	-
USER_ID	NOT NULL	NUMBER(5)	-
ACCOUNT_STATUS	-	VARCHAR2(4)	"OPEN"
LOCK_DATE	-	DATE	NULL

Table C–34 (Cont.) USER_USERS

Name	Null?	Type	Value
EXPIRY_DATE	-	DATE	NULL
DEFAULT_TABLESPACE	-	VARCHAR2(1)	NULL
TEMPORARY_TABLESPACE	-	VARCHAR2(1)	NULL
CREATED	NOT NULL	DATE	-
INITIAL_RSRC_CONSUMER_GROUP	-	VARCHAR2(1)	NULL
EXTERNAL_NAME	-	VARCHAR2(1)	NULL

Table C–35 USER_VIEWS

Name	Null?	Type	Value
VIEW_NAME	-	VARCHAR2(256)	-
TEXT_LENGTH	-	NUMBER	0
TEXT	-	VARCHAR2(256)	-
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

D Heterogeneous Services Initialization Parameters

The Oracle database server initialization parameters in the init.ora file are distinct from heterogeneous services (HS) initialization parameters. Set HS parameters in the initialization parameter file using an agent-specific mechanism, or set them in the Oracle data dictionary using the DBMS_HS package.

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

- HS_FDS_CONNECT_INFO
- HS_FDS_DEFAULT_OWNER
- HS_FDS_PARSER_TOKEN_SIZE
- HS_FDS_PROC_IS_FUNC
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_REPORT_REAL_AS_DOUBLE
- HS_FDS_RESULTSET_SUPPORT
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_TRANSACTION_MODEL

> **See Also:** *Oracle Database Heterogeneous Connectivity Administrator's Guide* for information on other available initialization parameters.

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

```
ORACLE_HOME\tg4msql\admin\inittg4msql.ora
```

Where ORACLE_HOME is the directory under which the gateway is installed.

HS 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. The keywords '`set`' and '`private`' are reserved. 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 new line
      * \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	Range of Values
None	Not Applicable

Specifies the information needed to connect to the Microsoft SQL Server 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 Microsoft SQL Server data and *database_name* is the name of the database.

If you specify only *database_name*, omitting *server_name* (but including the period), the gateway uses the following subtree of the Microsoft Windows registry to determine *server_name*:

HKEY_LOCAL_MACHINE\

SOFTWARE\

Microsoft\

MSSQLServer\

Client\

ConnectTo

HS_FDS_DEFAULT_OWNER

Default Value	Range of Values
Not Applicable	Any value

The name of the table owner that is used for the Microsoft SQL Server 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_PARSER_TOKEN_SIZE

Default Value	Range of Values
1,000 Characters	Any positive integer value

Used for setting the parser token size in case the default size is not sufficient. The default value can be changed in cases when the following error occurs:

```
pclex input buffer overflowed, try to increase the variable tokenSize in your
evironment.
```

With default value of 1000, the gateway could handle SQL statements close to 2M. Note that the SQL statements sent to the gateway could be very different from the SQL statements issued by the users. If in doubt, turn on gateway trace. Increase this parameter to handle larger SQL statements sent to gateways.

HS_FDS_PROC_IS_FUNC

Default Value	Range of Values
FALSE	TRUE, FALSE

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 to handle return values.

HS_FDS_RECOVERY_ACCOUNT

Default Value	Range of Values
RECOVER	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 Microsoft SQL Server. For more information about the commit-confirm model, see the HS_FDS_TRANSACTION_MODEL parameter.

HS_FDS_RECOVERY_PWD

Default Value	Range of Values
RECOVER	Any valid password

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

HS_FDS_REPORT_REAL_AS_DOUBLE

Default Value	Range of Values
FALSE	TRUE, FALSE

Enables Microsoft SQL gateway to treat SINGLE FLOAT PERCISION fields as DOUBLE FLOAT PERCISION fields.

HS_FDS_RESULTSET_SUPPORT

Default Value	Range of Values
FALSE	TRUE, FALSE

Enables result sets to be returned from stored procedures. By default, all stored procedures do not return a result set to the user.

Note: If you set this initialization parameter, you must do the following:

- Change the syntax of the procedure execute statement for all existing stored procedures, to handle result sets
- Work in the sequential mode of Heterogeneous Services

HS_FDS_TRACE_LEVEL

Default Value	Range of Values
OFF	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\tg4msql\trace`.

HS_FDS_TRANSACTION_LOG

Default Value	Range of Values
HS_TRANSACTION_LOG	Any valid table name

Specifies the name of the table created in the Microsoft SQL Server database for logging transactions. For more information about the transaction model, see the HS_FDS_TRANSACTION_MODEL parameter.

HS_FDS_TRANSACTION_MODEL

Default Value	Range of Values
COMMIT_CONFIRM	COMMIT_CONFIRM, READ_ONLY, SINGLE_SITE or TWO_PHASE_COMMIT

Specifies the type of transaction model that is used when the Microsoft SQL Server database is updated by a transaction.

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the Microsoft SQL Server 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 Microsoft SQL Server database:
  - Transaction log table. The default table name is HS_TRANSACTION_LOG. A different name can be set using the HS_FDS_TRANSACTION_LOG parameter. 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 Microsoft SQL Server database.
- SINGLE_SITE provides read and write access to the Microsoft SQL Server database. However, the gateway cannot participate in distributed updates.
- TWO_PHASE_COMMIT provides read and write access to the Microsoft SQL Server database and allows the gateway to be part of a distributed update using the two phase commit protocol.

Index

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W