Oracle® Transparent Gateway for Teradata

Administrator's Guide

10*g* Release 2 (10.2) for HP-UX

B14276-01

June 2005

Oracle Transparent Gateway for Teradata Administrator's Guide, 10*g* Release 2 (10.2) for HP-UX

B14276-01

Primary Author: Amitai Sela

Contributing Author: Laurel Hale, Cynthia Kibbe, Kishan Peyetti

Contributor: Orit Curiel, Jacco Draaijer, Vira Goorah

Contents

3 Teradata Gateway Features and Restrictions

4 Case Studies

Index

List of Tables

Send Us Your Comments

Oracle Transparent Gateway for Teradata Administrator's Guide, 10*g* Release 2 (10.2) for HP-UX

B14276-01

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 Teradata, which enables Oracle client applications to access Teradata 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 Teradata
- Diagnosing gateway errors
- Using the gateway to access Teradata data

> **Note:** You should understand the fundamentals of transparent gateways and the HP-UX 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.

Related Documents

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 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 Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Conventions

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

- Conventions in Text

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.

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 Teradata 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 the *Oracle 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 that 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 Teradata.

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 must be installed on a machine running either the Teradata database or the Teradata 32-bit client. This machine can be the same machine as the Oracle database or on the same machine as the Teradata database or on a third machine as a standalone. Each configuration has its advantages and disadvantages. The issues to consider when determining where to install the gateway are network traffic, operating system platform availability, hardware resources and storage.

2

Configuring the Gateway

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

- Configuring the Gateway
- Configuring Oracle Net Services Listener for the Gateway
- Configuring the Oracle Database Server for Gateway Access
- Creating Database Links
- Configuring the Gateway for Multiple Teradata Databases
- Performing Configuration Tasks

Configuring the Gateway

Perform the following tasks to configure the Oracle Transparent Gateway for Teradata Administrator's Guide.

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 Teradata database you are accessing. The SID is used as part of the file name for the initialization parameter file. The default SID is `tg4tera`.

You can define a gateway SID, but using the default of `tg4tera` is easier because you do not need to change the initialization parameter file name. However, if you want to access two Teradata databases, you need two gateway SIDs, one for each instance of the gateway. If you have one Teradata 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 Teradata 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/tg4tera/admin/inittg4tera.ora
```

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

If you are not using `tg4tera` 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 Teradata connection as follows:

```
HS_FDS_CONNECT_INFO=dsn
HS_FDS_SHAREABLE_NAME=odbc_installation_path/lib/libodbc.so
set ODBCINI = full_path_to_the_odbcini_file
```

Where:

- *dsn* is the Data Source Name defined in the odbc.ini file.
- *full_path_to_the_odbcini_file* is the full path location of the odbc initialization file.
- *odbc_installation_path* is the path where the NCR Teradata ODBC Driver is installed.

  Note that the set statement is optional as long as it is specified in the working account.

> **Note:** In the ODBC DSN set up, make sure that the DateTimeFormat parameter is set to AAA

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

Example of Gateway to Start in listener.ora File

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

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (SID_NAME=gateway_sid)
         (ORACLE_HOME=oracle_home_directory)
         (PROGRAM=tg4tera)
         (ENVS=SHLIB_PATH=odbc_installation_path/lib:
                                          oracle_home_directory/lib32)
      )
   )
```

Table 2–1 defines the parameters in the entry.

Table 2–1 SID_LIST_LISTENER Parameters

Variable	Description
gateway_sid	Specifies the SID of the gateway. 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.
`tg4tera`	Specifies the Oracle Transparent Gateway for Teradata Administrator's Guide.
odbc_installation_path	Specifies the Oracle Transparent Gateway for Teradata Administrator's Guide.

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=tg4tera)

(ENVS=SHLIB_PATH=odbc_installation_path/lib:
                                        oracle_home_directory/lib32)
  )
)
```

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

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

   If you have the C Shell, enter the following:

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

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

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

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

   ```
   $ lsnrctl status
   ```

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

```
LSNRCTL for HPUX: Version 10.2.0.1.0 - Production on 01-JUN-2005 09:28:13

Copyright (c) 1991, 2004, Oracle. All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=204.179.99.15)(PORT=1551))
STATUS of the LISTENER
----------------------
Alias                     listener
Version                   TNSLSNR for HPUX: Version 10.2.0.1.0 - Production
Start Date                28-APRIL-2005 15:38:59
Uptime                    33 days 17 hr. 49 min. 22 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File   /users/oracle/gateway/network/admin/listener.ora
Listener Log File         /users/oracle/gateway/network/log/listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=204.179.99.15)(PORT=1551)))
Services Summary...
```

```
Service "tg4tera" has 1 instance(s).
  Instance "tg4tera", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

In this example, `tg4tera` is the default SID value assigned during installation. You can use any valid ID for the SID, or keep the default.

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

Configuring the Oracle Database Server for Gateway Access

Before you use the gateway to access Teradata 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))
```

Table 2–2 on page 2-6 defines the parameters used in the connect descriptor.

Table 2–2 Parameters for Gateway Connect Descriptor

Variable	Description
connect_descriptor	The description of the object to connect to as specified when creating the database link, such as `tg4tera`. Check the `sqlnet.ora` file for the following parameter setting: `names.directory_path = (TNSNAMES)` **Note:** The `sqlnet.ora` file is typically stored in `$ORACLE_HOME/network/admin`.
`TCP`	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 2-3.
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 Teradata 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 Teradata 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 Teradata database.

> **See Also:** *Oracle Database Administrator's Guide* and *Oracle Database Heterogeneous Services 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 Teradata Databases

The tasks for configuring the gateway to access multiple Teradata 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 `tg4tera`
- The *ORACLE_HOME* environment variable is set to the directory where the gateway is installed.
- The gateway is configured for one Teradata database named db1.
- Two Teradata databases named db2 and db3 on a server are being added.

Configuring the gateway for additional Teradata 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 Teradata Database

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

- `tg4tera2` for the gateway accessing database db2.
- `tg4tera3` 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/tg4tera/admin/inittg4tera.ora`, twice, naming one with the gateway SID for db2 and the other with the gateway SID for db3:

```
$ cd $ORACLE_HOME/tg4tera/admin
$ cp inittg4tera.ora inittg4tera2.ora
$ cp inittg4tera.ora inittg4tera3.ora
```

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

For `inittg4tera2.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=dsn2
```

For `inittg4tera3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=dsn3
```

Note: If you have multiple gateway SIDs for the same Teradata 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=tg4tera)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4tera)

(ENVS=SHLIB_PATH=odbc_installation_path/lib:
oracle_home_directory/lib32)
   )
   (SID_DESC=
      (SID_NAME=tg4tera2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4tera)

(ENVS=SHLIB_PATH=odbc_installation_path/lib:
oracle_home_directory/lib32)
   )
   (SID_DESC=
      (SID_NAME=tg4tera3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4tera)

(ENVS=SHLIB_PATH=odbc_installation_path/lib:
oracle_home_directory/lib32)
   )
)
```

Multiple Databases Example: Stopping and Starting the TNS Listener

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

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

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

Configuring 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 Teradata 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=tg4tera))
               (HS=OK))
new_db2_using=(DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                    (SID=tg4tera2))
                (HS=OK))
new_db3_using=(DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                    (SID=tg4tera3))
                (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 Teradata Data

Enter the following to create a database link for the `tg4tera2` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK TERA2 CONNECT TO
  2  user2 IDENTIFIED BY password2 USING 'new_db2_using';
```

Enter the following to create a database link for the `tg4tera3` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK TERA3 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 Teradata databases, as in the following:

```
SQL> SELECT * FROM ALL_USERS@TERA2;
```

Or

```
SQL> SELECT * FROM ALL_USERS@TERA3;
```

Performing Configuration Tasks

You can perform the following configuration tasks:

- Configuring for Two-Phase Commit

Configuring for Two-Phase Commit

The gateway supports the following transaction capabilities:

- COMMIT_CONFIRM
- READ_ONLY
- SINGLE_SITE

By default, the gateway runs in COMMIT_CONFIRM transaction mode. When the Teradata database is updated by a transaction, the gateway becomes the commit point site. The Oracle database server commits the unit of work in the Teradata 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 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 Teradata 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 Teradata database. Both the user name and password must be a valid Teradata 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 Teradata user account you set up for recovery.
   - HS_FDS_RECOVERY_PWD to the password of the Teradata 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 Teradata database for logging transactions. The gateway uses the transaction log table to check the status of failed transactions that were started at the Teradata 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) 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 2-10. Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database where the Teradata 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 `tg4tera_tx.sql` script, located in the directory `$ORACLE_HOME/tg4tera/admin`, where *$ORACLE_HOME* is the directory under which the gateway is installed.

3

Teradata Gateway Features and Restrictions

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

This chapter contains the following sections:

Using the Pass-Through Feature

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

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

Where *command* cannot be one of the following:

- BEGIN TRANSACTION
- BT
- COMMIT
- END TRANSACTION
- ET
- ROLLBACK

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

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

Database Compatibility Issues for Teradata

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

- Schema Considerations
- Naming Rules
- Data Types
- Queries
- Locking

Schema Considerations

The Oracle concept of a schema does not exist in Teradata. An owner included in a query is recognized by Teradata as a database name.

When querying data dictionary tables, the following results are returned:

- ALL_* data dictionary tables, data for every Teradata database is returned. The Teradata database name is returned in the owner column.
- ALL_USERS data dictionary table, the name of every Teradata database is returned.
- USER_* data dictionary tables, data for the default Teradata database is returned. If a default Teradata database is not defined, the DBC Teradata system database is used.

Note: You can specify a default database in Teradata in the DSN. For details refer to Teradata documentation.

Naming Rules

Naming rule issues include the following:

- Rules for Naming Objects
- Case Sensitivity

Rules for Naming Objects

Oracle and Teradata 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 Teradata documentation.

Case Sensitivity

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

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

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

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

If the Teradata 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"@TERA;
```

Or

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

Oracle Corporation recommends that you surround all Teradata object names with double quote characters and use the exact letter case for the object names as they appear in the Teradata 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 Teradata names to the correct letter case. For example, to refer to the Teradata 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"@TERA;
```

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

Data Types

Data type issues include the following:

- Binary Literal Notation
- Date Format Support
- 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 Teradata VARBINARY and BINARY data types (a *ff* surrounded by single quotes followed by hexadecimal digits).

For example, the following statement is not supported:

```
SQL> INSERT INTO BINARY_TAB@TERA VALUES ('ff'xb)
```

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.

Date Format Support

In order for Teradata date formats (DATE, TIME and TIMESTAMP) to be treated as dates and not strings, in the ODBC DSN set up make sure that the DateTimeFormat parameter is set to AAA.

Data Type Conversion

Teradata 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@
TERA
 WHERE DATE_COL = "1-JAN-2001";
```

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

```
SELECT DATE_COL FROM TEST@
TERA
 WHERE DATE_COL = TO_DATE("1-JAN-2001")
```

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

Queries

Query issues include the following:

- Row Selection
- Empty Bind Variables

Row Selection

Teradata 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 Bind Variables

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

Locking

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

Known Restrictions

If you encounter incompatibility problems not listed in this section or in "Known Problems" on page 3-7, please contact Oracle Support Services. The following section describes the known restrictions and includes suggestions for dealing with them when possible:

- Transactional Integrity
- Transaction Capability
- COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors
- Pass-Through Feature
- Bind Variables for Date Columns
- SQL Syntax
- SQL*Plus
- Database Links
- Stored Procedures

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

Transactional Integrity

The gateway cannot guarantee transactional integrity in the following cases:

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

Note: Oracle corporation strongly recommends the following:

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

Transaction Capability

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

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

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

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.

Pass-Through Feature

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

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

Bind Variables for Date Columns

Due to limitations in NCR's ODBC driver for Teradata, you cannot compare columns of data type TIME or TIMESTAMP to a bind variable.

The following SQL statement causes an error message:

```
SQL> select time_column from time_table@tg4tera where time_column = :a;
```

The following error is issued:

```
ORA-28500: connection from ORACLE to a non-Oracle system returned this

message:
[Transparent gateway for TERA]DRV_Execute: [NCR][ODBC Teradata][Teradata

RDBMS]
Invalid operation on an ANSI Datetime or Interval value. (SQL State: 22005;SQL
Code: -5407)
ORA-02063: preceding 2 lines from TERA
```

SQL Syntax

This section lists restrictions on the following SQL syntax:

- WHERE CURRENT OF Clause
- CONNECT BY Clause
- ROWID
- 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.

ROWID

The Oracle ROWID implementation is not supported.

EXPLAIN PLAN Statement

The EXPLAIN PLAN statement is not supported.

SQL*Plus

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 a shared server process and cannot support shared database links. Each gateway session spawns a separate gateway process and connections cannot be shared.

Stored Procedures

The gateway does not support the procedure feature that allows the execution of stored procedures in a non-Oracle database.

Known Problems

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

The following known problems are described in this section:

- Encrypted Format Login
- Teradata LONG VARCHAR Data Type
- Schema Names and PL/SQL
- Data Dictionary Views and PL/SQL

Encrypted Format Login

The Oracle database server no longer supports the initialization parameter DBLINK_ENCRYPT_LOGIN. Up to version 7.3, this parameter's default TRUE value prevented the password for the login user ID from being sent over the network (in the clear). Later versions automatically encrypt the password.

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.

Teradata LONG VARCHAR Data Type

The following restrictions apply when using LONG VARCHAR data types:

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

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

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

Schema Names and PL/SQL

If you do not prefix a Teradata 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.

4

Case Studies

The following case studies for Teradata 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 Teradata tables (Case 3)
- Write capabilities (Case 4)
- A data dictionary query (Case 5)
- The pass-through feature (Case 6)

CD-ROM Contents

The distribution CD-ROM contains the following:

- Demonstration files
- One SQL script file that creates the demonstration tables in the Teradata database
- One SQL script file that drops the demonstration tables from the Teradata database

Demonstration Files

After a successful gateway installation, use the demonstration files stored in the directory `$ORACLE_HOME/tg4tera/demo` where *$ORACLE_HOME* is the directory under which the gateway is installed. The directory contains the demonstration files listed in the following table:

Demonstration Files	Demonstration Files
bldtera.sql	case4c.sql
case1.sql	case5.sql
case2.sql	case6a.sql
case3.sql	case6b.sql
case4a.sql	droptera.sql
case4b.sql	-

Demonstration Requirements

The case studies assume these requirements have been met:

- The gateway demonstration tables are installed in the Teradata 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 Teradata 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. If the demonstration tables have not been created in the Teradata database, use the `bldtera.sql` script to create them.

The script creates the demonstration tables in the Teradata 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)

CREATE TABLE GTW_DEPT (
DEPTNO     SMALLINT NOT NULL,
DNAME      VARCHAR(14),
LOC        VARCHAR(13))

CREATE TABLE GTW_SALGRADE (
GRADE      REAL,
LOSAL      NUMERIC(9,4),
HISAL      NUMERIC(9,4))
```

Demonstration Table Definitions

The table definitions are listed here using 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                                   FLOAT(49)
LOSAL                                   NUMBER(9,4)
HISAL                                   NUMBER(9,4)
```

Demonstration Table Contents

The contents of the Teradata 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 Teradata Tables

Case 3 demonstrates the following:

- Joins between Teradata tables
- Subselects

The query retrieves information from three Teradata 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 Teradata 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 Teradata.

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.

A

Data Type Conversion

This appendix contains the following section:

- Data Type Conversion

Data Type Conversion

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

Table A–1 Data Type Conversions

Teradata	Oracle	Comment
BYTE	RAW	-
BYTEINT	NUMBER(3)	-
CHAR	CHAR	-
CHARACTER	CHAR	-
DATE	CHAR(10) or DATE	The conversion depends on the DateTimeFormat parameter in the ODBC DSN. Please refer to NCR's ODBC documentation
DEC	NUMBER(p[,s])	-
DECIMAL	NUMBER(p[,s])	-
DOUBLE PRECISION	FLOAT(49)	-
FLOAT	FLOAT(49)	-
INT	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
INTEGER	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
INTERVAL	Not supported	-
LONG VARCHAR	LONG	-
NUMERIC	NUMBER(p[,s])	-
REAL	FLOAT(49)	-
SMALLINT	NUMBER(5)	-
TIME	CHAR(8) or DATE	The conversion depends on the DateTimeFormat parameter in the ODBC DSN. Please refer to NCR's ODBC documentation
TIMESTAMP	CHAR(23) or DATE	The conversion depends on the DateTimeFormat parameter in the ODBC DSN. Please refer to NCR's ODBC documentation

Table A–1 (Cont.) Data Type Conversions

Teradata	Oracle	Comment
VARBYTE	RAW	VARBYTE is right padded. If the size is greater than 2000 it is truncated to RAW(2000)
VARCHAR	VARCHAR2	If the size is greater than 4000 it is truncated to VARCHAR2(4000)

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 Teradata database.

> **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 Teradata can be used.

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

INSERT

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

> **See Also:** "Functions Supported by Teradata" 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 Teradata can be used. Also, you cannot have SQL statements in the subquery that refer to the same table name in the outer query. Subqueries are not supported in the SET clause.

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

Oracle Functions

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

Functions Not Supported by Teradata

Oracle SQL functions with no equivalent function in Teradata 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 Teradata

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

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

Arithmetic Operators

Oracle	Teradata
+	+
-	-
*	*
/	/

Comparison Operators

Oracle	Teradata
=	=
>	>

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

Group Functions

Oracle	Teradata
AVG	AVG
COUNT	COUNT
MAX	MAX
MIN	MIN
SUM	SUM

String Functions

Oracle	Teradata
||	CONCAT
INSTR	LOCATE
LENGTH	LENGTH
SUBSTR (second argument cannot be a negative number)	SUBSTRING
UPPER	UCASE

Other Functions

Oracle	Teradata
ABS	ABS
EXP	EXP
LOG(a, b)	LOG10(b) / LOG10(a)
POWER	**
SQRT	SQRT

C

Data Dictionary

The Oracle Transparent Gateway for Teradata translates a query that refers to an Oracle database server data dictionary table into a query that retrieves the data from the Teradata system using ODBC Metadata APIs. 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 Teradata data dictionary support.

Teradata System Catalog Tables

Teradata data dictionary information is stored in the Teradata database as Teradata system catalog tables. The Teradata system catalog tables define the structure of a database. When you change data definitions, Teradata reads and modifies the Teradata system catalog 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 Teradata database:

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

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

1. Uses ODBC API to retrieve information from Teradata system catalog. Refere to "Data Dictionary Mapping" on page C-3 for details.
2. Sends the translated query to Teradata.

3. Might convert the retrieved Teradata data to give it the appearance of the Oracle database server data dictionary table.
4. Passes the data dictionary information from the translated Teradata system catalog table to the Oracle database server.

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

Direct Queries to Teradata Tables

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

Supported Views and Tables

The gateway supports the following views and tables:

Supported Views and Tables	Supported Views and Tables
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_TABLES
DBA_TAB_COLUMNS	DBA_TAB_COMMENTS
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 Teradata database even when access privileges for those Teradata 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

Oracle Transparent Gateway for Teradata uses ODBC Metadata API's to get all data dictionary information.

The tables in this section list Oracle data dictionary view names and ODBC API's used.

Table C–1 Oracle Data Dictionary View Names and Teradata Equivalents

View Name	ODBC API Used
ALL_CATALOG	SQLTables
ALL_COL_COMMENTS	SQLColumns
ALL_CONS_COLUMNS	SQLPrimaryKeys, SQLForeignKeys
ALL_CONSTRAINTS	SQLPrimaryKeys, SQLForeignKeys
ALL_IND_COLUMNS	SQLStatistics
ALL_INDEXES	SQLStatistics
ALL_OBJECTS	SQLTables, SQLProcedures, SQLStatistics
ALL_TAB_COLUMNS	SQLColumns
ALL_TAB_COMMENTS	SQLTables
ALL_TABLES	SQLStatistics
ALL_USERS	SQLTables
ALL_VIEWS	SQLTables
DBA_CATALOG	SQLTables
DBA_COL_COMMENTS	SQLColumns
DBA_OBJECTS	SQLTables, SQLProcedures, SQLStatistics
DBA_TAB_COLUMNS	SQLColumns
DBA_TAB_COMMENTS	SQLTables
DBA_TABLES	SQLStatistics
DICT_COLUMNS	SQLColumns
DICTIONARY	SQLTables
DUAL	*(Defined in the Gateway)*
USER_CATALOG	SQLTables
USER_COL_COMMENTS	SQLColumns
USER_CONS_COLUMNS	SQLPrimaryKeys, SQLForeignKeys
USER_CONSTRAINTS	SQLPrimaryKeys, SQLForeignKeys
USER_IND_COLUMNS	SQLStatistics
USER_INDEXES	SQLStatistics
USER_OBJECTS	SQLTables, SQLProcedures, SQLStatistics
USER_TAB_COLUMNS	SQLColumns

Table C–1 (Cont.) Oracle Data Dictionary View Names and Teradata Equivalents

View Name	ODBC API Used
USER_TAB_COMMENTS	SQLTables
USER_TABLES	SQLStatistics
USER_USERS	SQLTables
USER_VIEWS	SQLTables

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 Teradata system catalog 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 SQL*PLUS DESCRIBE command. The values in the Null? column might differ from the Oracle database server data dictionary tables and views. Any hardcoded default value is shown to the right of an item, but this is not information returned by DESCRIBE.

Table C–2 ALL_CATALOG

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"

Table C–3 ALL_COL_COMMENTS

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

Table C–4 ALL_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)	-

Table C–4 (Cont.) ALL_CONS_COLUMNS

Name	Null?	Type	Value
POSITION	-	FLOAT(49)	-

Table C–5 ALL_CONSTRAINTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)	-
CONSTRAINT_TYPE	-	VARCHAR2(1)	"R" or "P"
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
SEARCH_CONDITION	NOT NULL	CHAR(1)	" "
R_OWNER	NOT NULL	VARCHAR2(30)	" "
R_CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)	" "
DELETE_RULE	NOT NULL	VARCHAR2(9)	" "
STATUS	NOT NULL	CHAR(1)	" "
DEFERRABLE	NOT NULL	CHAR(1)	" "
DEFERRED	NOT NULL	CHAR(1)	" "
VALIDATED	NOT NULL	CHAR(1)	" "
GENERATED	NOT NULL	CHAR(1)	" "
BAD	NOT NULL	CHAR(1)	" "
RELY	NOT NULL	CHAR(1)	" "
LAST_CHANGE	-	DATE	NULL

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)	-
COLUMN_LENGTH	NOT NULL	FLOAT(49)	-
DESCEND	-	VARCHAR2(4)	"DESC" or "ASC"

Table C–7 ALL_INDEXES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
INDEX_NAME	NOT NULL	CHAR(1)	-
INDEX_TYPE	NOT NULL	CHAR(1)	" "

Table C–7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
TABLE_OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	NOT NULL	CHAR(5)	"TABLE"
UNIQUENESS	-	VARCHAR2(9)	"UNIQUE" or "NONUNIQUE"
COMPRESSION	NOT NULL	CHAR(1)	" "
PREFIX_LENGTH	NOT NULL	NUMBER(10)	0
TABLESPACE_NAME	NOT NULL	CHAR(1)	" "
INI_TRANS	NOT NULL	NUMBER(10)	0
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
PCT_THRESHOLD	NOT NULL	NUMBER(10)	0
INCLUDE_COLUMNS	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
PCT_FREE	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "
BLEVEL	NOT NULL	NUMBER(10)	0
LEAF_BLOCKS	NOT NULL	NUMBER(10)	0
DISTINCT_KEYS	-	FLOAT(49)	-
AVG_LEAF_BLOCKS_PER_KEY	NOT NULL	NUMBER(10)	0
AVG_DATA_BLOCKS_PER_KEY	NOT NULL	NUMBER(10)	0
CLUSTERING_FACTOR	NOT NULL	NUMBER(10)	0
STATUS	NOT NULL	CHAR(1)	" "
NUM_ROWS	NOT NULL	NUMBER(10)	0
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
DEGREE	NOT NULL	CHAR(1)	" "
INSTANCES	NOT NULL	CHAR(1)	" "
PARTITIONED	NOT NULL	CHAR(1)	" "
TEMPORARY	NOT NULL	CHAR(1)	" "
GENERATED	NOT NULL	CHAR(1)	" "
SECONDARY	NOT NULL	CHAR(1)	" "

Table C–7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
BUFFER_POOL	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "
PCT_DIRECT_ACCESS	NOT NULL	NUMBER(10)	0
ITYP_OWNER	NOT NULL	CHAR(1)	" "
ITYP_NAME	NOT NULL	CHAR(1)	" "
PARAMETERS	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
DOMIDX_STATUS	NOT NULL	CHAR(1)	" "
DOMIDX_OPSTATUS	NOT NULL	CHAR(1)	" "
FUNCIDX_STATUS	NOT NULL	CHAR(1)	" "

Table C–8 ALL_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(30)	-
OBJECT_NAME	-	VARCHAR2(30)	-
SUBOBJECT_NAME	-	VARCHAR2(1)	NULL
OBJECT_ID	-	NUMBER	0
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2(18)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	NULL
LAST_DDL_TIME	-	DATE	NULL
TIMESTAMP	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL

Table C–9 ALL_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
COLUMN_NAME	NOT NULL	VARCHAR2(30)	-
DATA_TYPE	-	VARCHAR2(106)	-
DATA_TYPE_MOD	NOT NULL	CHAR(1)	" "

Table C–9 (Cont.) ALL_TAB_COLUMNS

Name	Null?	Type	Value
DATA_TYPE_OWNER	NOT NULL	CHAR(1)	" "
DATA_LENGTH	NOT NULL	FLOAT(49)	-
DATA_PRECISION	-	FLOAT(49)	-
DATA_SCALE	-	FLOAT(49)	-
NULLABLE	-	VARCHAR2(1)	"Y" or "N"
COLUMN_ID	NOT NULL	FLOAT(49)	-
DEFAULT_LENGTH	NOT NULL	NUMBER(10)	0
DATA_DEFAULT	NOT NULL	CHAR(1)	" "
NUM_DISTINCT	NOT NULL	NUMBER(10)	0
LOW_VALUE	NOT NULL	NUMBER(10)	0
HIGH_VALUE	NOT NULL	NUMBER(10)	0
DENSITY	NOT NULL	NUMBER(10)	0
NUM_NULLS	NOT NULL	NUMBER(10)	0
NUM_BUCKETS	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
CHARACTER_SET_NAME	NOT NULL	CHAR(1)	" "
CHAR_COL_DECL_LENGTH	NOT NULL	NUMBER(10)	0
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
AVG_COL_LEN	NOT NULL	NUMBER(10)	0

Table C–10 ALL_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–11 ALL_TABLES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLESPACE_NAME	NOT NULL	CHAR(1)	" "
CLUSTER_NAME	NOT NULL	CHAR(1)	" "
IOT_NAME	NOT NULL	CHAR(1)	" "

Table C–11 (Cont.) ALL_TABLES

Name	Null?	Type	Value
PCT_FREE	NOT NULL	NUMBER(10)	0
PCT_USED	NOT NULL	NUMBER(10)	0
INI_TRANS	NOT NULL	NUMBER(10)	0
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "
BACKED_UP	NOT NULL	CHAR(1)	" "
NUM_ROWS	-	FLOAT(49)	-
BLOCKS	-	FLOAT(49)	-
EMPTY_BLOCKS	NOT NULL	NUMBER(10)	0
AVG_SPACE	NOT NULL	NUMBER(10)	0
CHAIN_CNT	NOT NULL	NUMBER(10)	0
AVG_ROW_LEN	NOT NULL	NUMBER(10)	0
AVG_SPACE_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
NUM_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
DEGREE	NOT NULL	CHAR(1)	" "
INSTANCES	NOT NULL	CHAR(1)	" "
CACHE	NOT NULL	CHAR(1)	" "
TABLE_LOCK	NOT NULL	CHAR(1)	" "
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
PARTITIONED	NOT NULL	CHAR(1)	" "
IOT_TYPE	NOT NULL	CHAR(1)	" "
TEMPORARY	NOT NULL	CHAR(1)	" "
SECONDARY	NOT NULL	CHAR(1)	" "
NESTED	NOT NULL	CHAR(1)	" "
BUFFER_POOL	NOT NULL	CHAR(1)	" "
ROW_MOVEMENT	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "

Table C–11 (Cont.) ALL_TABLES

Name	Null?	Type	Value
SKIP_CORRUPT	NOT NULL	CHAR(1)	" "
MONITORING	NOT NULL	CHAR(1)	" "

Table C–12 ALL_USERS

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

Table C–13 ALL_VIEWS

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

Table C–14 DBA_CATALOG

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"

Table C–15 DBA_COL_COMMENTS

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

Table C–16 DBA_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(30)	-
OBJECT_NAME	-	VARCHAR2(128)	-
SUBOBJECT_NAME	-	VARCHAR2(1)	NULL
OBJECT_ID	-	NUMBER	0
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2(18)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	NULL
LAST_DDL_TIME	-	DATE	NULL
TIMESTAMP	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL

Table C–17 DBA_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
COLUMN_NAME	NOT NULL	VARCHAR2(30)	-
DATA_TYPE	-	VARCHAR2(106)	-
DATA_TYPE_MOD	NOT NULL	CHAR(1)	" "
DATA_TYPE_OWNER	NOT NULL	CHAR(1)	" "
DATA_LENGTH	NOT NULL	FLOAT(49)	-
DATA_PRECISION	-	FLOAT(49)	-
DATA_SCALE	-	FLOAT(49)	-
NULLABLE	-	VARCHAR2(1)	"Y" or "N"
COLUMN_ID	NOT NULL	FLOAT(49)	-
DEFAULT_LENGTH	NOT NULL	NUMBER(10)	0
DATA_DEFAULT	NOT NULL	CHAR(1)	" "
NUM_DISTINCT	NOT NULL	NUMBER(10)	0
LOW_VALUE	NOT NULL	NUMBER(10)	0
HIGH_VALUE	NOT NULL	NUMBER(10)	0
DENSITY	NOT NULL	NUMBER(10)	0
NUM_NULLS	NOT NULL	NUMBER(10)	0
NUM_BUCKETS	NOT NULL	NUMBER(10)	0

Table C–17 (Cont.) DBA_TAB_COLUMNS

Name	Null?	Type	Value
LAST_ANALYZED	-	DATE	NULL
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
CHARACTER_SET_NAME	NOT NULL	CHAR(1)	" "
CHAR_COL_DECL_LENGTH	NOT NULL	NUMBER(10)	0
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
AVG_COL_LEN	NOT NULL	NUMBER(10)	0

Table C–18 DBA_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–19 DBA_TABLES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLESPACE_NAME	NOT NULL	CHAR(1)	" "
CLUSTER_NAME	NOT NULL	CHAR(1)	" "
IOT_NAME	NOT NULL	CHAR(1)	" "
PCT_FREE	NOT NULL	NUMBER(10)	0
PCT_USED	NOT NULL	NUMBER(10)	0
INI_TRANS	NOT NULL	NUMBER(10)	0
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "
BACKED_UP	NOT NULL	CHAR(1)	" "
NUM_ROWS	-	FLOAT(49)	-

Table C–19 (Cont.) DBA_TABLES

Name	Null?	Type	Value
BLOCKS	-	FLOAT(49)	-
EMPTY_BLOCKS	NOT NULL	NUMBER(10)	0
AVG_SPACE	NOT NULL	NUMBER(10)	0
CHAIN_CNT	NOT NULL	NUMBER(10)	0
AVG_ROW_LEN	NOT NULL	NUMBER(10)	0
AVG_SPACE_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
NUM_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
DEGREE	NOT NULL	CHAR(1)	" "
INSTANCES	NOT NULL	CHAR(1)	" "
CACHE	NOT NULL	CHAR(1)	" "
TABLE_LOCK	NOT NULL	CHAR(1)	" "
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
PARTITIONED	NOT NULL	CHAR(1)	" "
IOT_TYPE	NOT NULL	CHAR(1)	" "
TEMPORARY	NOT NULL	CHAR(1)	" "
SECONDARY	NOT NULL	CHAR(1)	" "
NESTED	NOT NULL	CHAR(1)	" "
BUFFER_POOL	NOT NULL	CHAR(1)	" "
ROW_MOVEMENT	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "
SKIP_CORRUPT	NOT NULL	CHAR(1)	" "
MONITORING	NOT NULL	CHAR(1)	" "

Table C–20 DICT_COLUMNS

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

Table C–21 DICTIONARY

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

Table C–22 DUAL

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

Table C–23 USER_CATALOG

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"

Table C–24 USER_COL_COMMENTS

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

Table C–25 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–26 USER_CONSTRAINTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)	-
CONSTRAINT_TYPE	-	VARCHAR2(1)	"R" or "P"
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
SEARCH_CONDITION	NOT NULL	CHAR(1)	" "
R_OWNER	NOT NULL	VARCHAR2(30)	-
R_CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)	-
DELETE_RULE	NOT NULL	VARCHAR2(9)	" "
STATUS	NOT NULL	CHAR(1)	" "
DEFERRABLE	NOT NULL	CHAR(1)	" "
DEFERRED	NOT NULL	CHAR(1)	" "
VALIDATED	NOT NULL	CHAR(1)	" "
GENERATED	NOT NULL	CHAR(1)	" "
BAD	NOT NULL	CHAR(1)	" "

Table C–26 (Cont.) USER_CONSTRAINTS

Name	Null?	Type	Value
RELY	NOT NULL	CHAR(1)	" "
LAST_CHANGE	-	DATE	NULL

Table C–27 USER_IND_COLUMNS

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

Table C–28 USER_INDEXES

Name	Null?	Type	Value
INDEX_NAME	NOT NULL	VARCHAR2(30)	-
INDEX_TYPE	NOT NULL	CHAR(1)	" "
TABLE_OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	NOT NULL	CHAR(5)	"TABLE"
UNIQUENESS	-	VARCHAR2(9)	"UNIQUE" or "NONUNIQUE"
COMPRESSION	NOT NULL	CHAR(1)	" "
PREFIX_LENGTH	NOT NULL	NUMBER(10)	0
TABLESPACE_NAME	NOT NULL	CHAR(1)	" "
INI_TRANS	NOT NULL	NUMBER(10)	0
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
PCT_THRESHOLD	NOT NULL	NUMBER(10)	0
INCLUDE_COLUMNS	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
PCT_FREE	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "

Table C–28 (Cont.) USER_INDEXES

Name	Null?	Type	Value
BLEVEL	NOT NULL	NUMBER(10)	0
LEAF_BLOCKS	NOT NULL	NUMBER(10)	0
DISTINCT_KEYS	-	FLOAT(49)	-
AVG_LEAF_BLOCKS_PER_KEY	NOT NULL	NUMBER(10)	0
AVG_DATA_BLOCKS_PER_KEY	NOT NULL	NUMBER(10)	0
CLUSTERING_FACTOR	NOT NULL	NUMBER(10)	0
STATUS	NOT NULL	CHAR(1)	" "
NUM_ROWS	NOT NULL	NUMBER(10)	0
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
DEGREE	NOT NULL	CHAR(1)	" "
INSTANCES	NOT NULL	CHAR(1)	" "
PARTITIONED	NOT NULL	CHAR(1)	" "
TEMPORARY	NOT NULL	CHAR(1)	" "
GENERATED	NOT NULL	CHAR(1)	" "
SECONDARY	NOT NULL	CHAR(1)	" "
BUFFER_POOL	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "
PCT_DIRECT_ACCESS	NOT NULL	NUMBER(10)	0
ITYP_OWNER	NOT NULL	CHAR(1)	" "
ITYP_NAME	NOT NULL	CHAR(1)	" "
PARAMETERS	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
DOMIDX_STATUS	NOT NULL	CHAR(1)	" "
DOMIDX_OPSTATUS	NOT NULL	CHAR(1)	" "
FUNCIDX_STATUS	NOT NULL	CHAR(1)	" "

Table C–29 USER_OBJECTS

Name	Null?	Type	Value
OBJECT_NAME	-	VARCHAR2(128)	-
SUBOBJECT_NAME	-	VARCHAR2(1)	NULL
OBJECT_ID	-	NUMBER	0
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2(18)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"

Table C–29 (Cont.) USER_OBJECTS

Name	Null?	Type	Value
CREATED	-	DATE	NULL
LAST_DDL_TIME	-	DATE	NULL
TIMESTAMP	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL

Table C–30 USER_TAB_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
COLUMN_NAME	NOT NULL	VARCHAR2(30)	-
DATA_TYPE	-	VARCHAR2(106)	-
DATA_TYPE_MOD	NOT NULL	CHAR(1)	" "
DATA_TYPE_OWNER	NOT NULL	CHAR(1)	" "
DATA_LENGTH	NOT NULL	FLOAT(49)	-
DATA_PRECISION	-	FLOAT(49)	-
DATA_SCALE	-	FLOAT(49)	-
NULLABLE	-	VARCHAR2(1)	"Y" or "N"
COLUMN_ID	NOT NULL	FLOAT(49)	-
DEFAULT_LENGTH	NOT NULL	NUMBER(10)	0
DATA_DEFAULT	NOT NULL	CHAR(1)	" "
NUM_DISTINCT	NOT NULL	NUMBER(10)	0
LOW_VALUE	NOT NULL	NUMBER(10)	0
HIGH_VALUE	NOT NULL	NUMBER(10)	0
DENSITY	NOT NULL	NUMBER(10)	0
NUM_NULLS	NOT NULL	NUMBER(10)	0
NUM_BUCKETS	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
CHARACTER_SET_NAME	NOT NULL	CHAR(1)	" "
CHAR_COL_DECL_LENGTH	NOT NULL	NUMBER(10)	0
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
AVG_COL_LEN	NOT NULL	NUMBER(10)	0

Table C–31 USER_TAB_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–32 USER_TABLES

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLESPACE_NAME	NOT NULL	CHAR(1)	" "
CLUSTER_NAME	NOT NULL	CHAR(1)	" "
IOT_NAME	NOT NULL	CHAR(1)	" "
PCT_FREE	NOT NULL	NUMBER(10)	0
PCT_USED	NOT NULL	NUMBER(10)	0
INI_TRANS	NOT NULL	NUMBER(10)	0
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "
BACKED_UP	NOT NULL	CHAR(1)	" "
NUM_ROWS	-	FLOAT(49)	-
BLOCKS	NOT NULL	FLOAT(49)	-
EMPTY_BLOCKS	NOT NULL	NUMBER(10)	0
AVG_SPACE	NOT NULL	NUMBER(10)	0
CHAIN_CNT	NOT NULL	NUMBER(10)	0
AVG_ROW_LEN	NOT NULL	NUMBER(10)	0
AVG_SPACE_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
NUM_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
DEGREE	NOT NULL	CHAR(1)	" "
INSTANCES	NOT NULL	CHAR(1)	" "
CACHE	NOT NULL	CHAR(1)	" "
TABLE_LOCK	NOT NULL	CHAR(1)	" "
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0

Table C–32 (Cont.) USER_TABLES

Name	Null?	Type	Value
LAST_ANALYZED	-	DATE	NULL
PARTITIONED	NOT NULL	CHAR(1)	" "
IOT_TYPE	NOT NULL	CHAR(1)	" "
TEMPORARY	NOT NULL	CHAR(1)	" "
SECONDARY	NOT NULL	CHAR(1)	" "
NESTED	NOT NULL	CHAR(1)	" "
BUFFER_POOL	NOT NULL	CHAR(1)	" "
ROW_MOVEMENT	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "
SKIP_CORRUPT	NOT NULL	CHAR(1)	" "
MONITORING	NOT NULL	CHAR(1)	" "

Table C–33 USER_USERS

Name	Null?	Type	Value
USERNAME	NOT NULL	VARCHAR2(30)	-
USER_ID	-	NUMBER	0
ACCOUNT_STATUS	-	VARCHAR2(4)	"OPEN"
LOCK_DATE	-	DATE	NULL
EXPIRY_DATE	-	DATE	NULL
DEFAULT_TABLESPACE	-	VARCHAR2(1)	NULL
TEMPORARY_TABLESPACE	-	VARCHAR2(1)	NULL
CREATED	-	DATE	NULL
INITIAL_RSRC_CONSUMER_GROUP	-	VARCHAR2(1)	NULL
EXTERNAL_NAME	-	VARCHAR2(1)	NULL

Table C–34 USER_VIEWS

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

Table C–34 (Cont.) USER_VIEWS

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

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_PARSER_TOKEN_SIZE
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_SHAREABLE_NAME
- 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/tg4tera/admin/inittg4tera.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 Teradata database.

This is a required parameter, whose format is:

```
HS_FDS_CONNECT_INFO=dsn
```

Where:

- *dsn* is the Data Source Name defined in the odbc.ini file.

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_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 Teradata. For more information about the commit-confirm model, see the HS_FDS_TRANSACTION_MODEL parameter.

The name of the recovery account is case-sensitive.

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 Teradata. For more information about the commit-confirm model, see the HS_FDS_TRANSACTION_MODEL parameter.

The name of the password of the recovery account is case-sensitive.

HS_FDS_SHAREABLE_NAME

Default Value	Range of Values
None	Not applicable

Specifies the full path name to the ODBC library.

This is a required parameter, whose format is:

```
HS_FDS_SHAREABLE_NAME=odbc_installation_path/lib/libodbc.sl
```

Where:

- *odbc_installation_path* is the path where the NCR Teradata ODBC Driver is installed.

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/TG4TERA/LOG`.

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 Teradata 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, or SINGLE_SITE

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

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the Teradata 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 Teradata 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 Teradata database.
- SINGLE_SITE provides read and write access to the Teradata database. However, the gateway cannot participate in distributed updates.

Index

H

I

K

L

N

O

P

R

S

T

U

V

W