Oracle® Database Gateway

Installation and Configuration Guide

11*g* Release 1 (11.1) for Microsoft Windows

B31043-01

March 2007

Beta Draft

ORACLE®

Oracle Database Gateway Installation and Configuration Guide, 11*g* Release 1 (11.1) for Microsoft Windows

B31043-01

Primary Author: Maitreyee Chaliha

Contributing Author: Vira Goorah, Govind Lakkoju, Peter Wong

Contents

Preface

This guide describes how to install Oracle Database Gateway for Microsoft SQL Server, Sybase, Teradata, ODBC, and DRDA on Microsoft Windows (32-bit) platform.

Intended Audience

This manual is intended for users responsible for installing Oracle Database Gateway on Microsoft Windows (32-bit).

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 Gateway for Sybase User's Guide*
- *Oracle Database Gateway for Informix User's Guide*
- *Oracle Database Gateway for Teradata User's Guide*
- *Oracle Database Gateway for Microsoft SQL Server User's Guide*
-
- *Oracle Database Gateway for DRDA User's Guide*
- *Oracle Database New Features Guide*
- *Oracle Call Interface Programmer's Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database Advanced Application Developer's Guide*
- *Oracle Database Concepts*
- *Oracle Database Performance Tuning Guide*
- *Oracle Database Error Messages*
- *Oracle Database Globalization Support Guide*
- *Oracle Database Reference*
- *Oracle Database SQL Language Reference*
- *Oracle Database Net Services Administrator's Guide*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database Heterogeneous Connectivity Administrator's Guide*
- *Oracle Database Security Guide*

Conventions

The following text conventions are used in this manual:

Convention	Meaning
bold	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary
italics	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter, directory names, usernames, pathnames, and filenames.
UPPERCASE	Uppercase letters indicate Structured Query Language (SQL) reserved words, initialization parameters, and environment variables.
[text]	Brackets are used in syntax statements for optional elements.
[text | text]	Vertical bar inside brackets is used in syntax statements to imply choice among optional elements.

Convention	Meaning
{text | text}	Vertical bar inside braces is used in syntax statements to imply choice among mandatory elements.

Part I

Overview of the Oracle Database Gateway Installation

Part I contains the following chapter:

- Chapter 1, "Overview of the Oracle Database Gateway Installation"

1

Overview of the Oracle Database Gateway Installation

This chapter describes the installation of Oracle Database Gateways on Microsoft Windows (32-bit) as well as issues that you should consider before installing the software. It includes the following topics:

Gateway Installation Configurations

You can install the gateway in either of the following configurations:

1. On the same computer as an existing Oracle database but in a different Oracle home.
2. On a system with no Oracle database.
3. On the same computer as the Oracle database and in the same Oracle home directory. Note that in this case, the Oracle database and the gateway must be at the same release level.

Gateway Installation Methods

You can choose different installation methods to install the gateway, as follows:

Interactive Installation Method

When you use the interactive method to install the gateway, Oracle Universal Installer displays a series of screens that enable you to specify all of the required information to install the gateway software.

Automated Installation Method Using Response Files

By creating a response file and specifying this file when you start Oracle Universal Installer, you can automate some or all of the gateway installation.

For more information about these modes and about how to complete an installation using response files, refer to Appendix A, "Using Response Files for Noninteractive Installation".

Installation Considerations

This section contains information that you should consider before installing this product. It contains the following sections:

- Release Notes
- Hardware and Software Certification
- Multiple Oracle Homes

Release Notes

Read the release notes for the product before installing it. The release notes are available on the Oracle Database 11*g* Release 1 (11.1) installation media. The latest version of the release notes is also available on the Oracle Technology Network (OTN) Web site:

`http://www.oracle.com/technology/documentation/index.html`

Hardware and Software Certification

The platform-specific hardware and software requirements included in this installation guide were current at the time this guide was published. However, because new platforms and operating system software versions might be certified after this guide is published, review the certification matrix on the Oracle*MetaLink* Web site for the most up-to-date list of certified hardware platforms and operating system versions. The Oracle*MetaLink* Web site is available at the following Web site:

`http://metalink.oracle.com`

If you do not have a current Oracle Support Services contract, then you can access the same information at the following Web site:

`http://www.oracle.com/technology/support/metalink/content.html`

Multiple Oracle Homes

This product supports multiple Oracle homes. This means that you can install this release or previous releases of the software more than once on the same system, in different Oracle home directories.

Installing the Software on a System with an Existing Oracle Installation

You must install this product in a new Oracle home directory. You cannot install products from one release of the gateway into an Oracle home directory of a different release. For example, you cannot install release 10.1 software into an existing Oracle9i Oracle home directory. If you attempt to install this release in an Oracle home directory that contains software from an earlier Oracle release, then the installation will fail.

You can install this release more than once on the same system if each installation is installed in a separate Oracle home directory.

Upgrades

Upgrades are not supported for the gateway.

Accessing the Installation Software

You can access and install the gateway by using the following methods:

- Downloading Oracle Software from the Oracle Technology Network Web Site
- Copying the Gateway Software to a Hard Disk

Downloading Oracle Software from the Oracle Technology Network Web Site

You can download the installation files from the OTN and extract them to your hard disk.

To download the installation files:

1. Use any browser to access the OTN software download page:

   `http://www.oracle.com/technology/software/index.html`

2. Navigate to each of the download pages for the product that you want to install.
3. On each download page, identify the required disk space by adding the file sizes for each required file. The file sizes are listed next to the file names.
4. Select a file system with enough free space to store and expand the files. In most cases, the available disk space must be at least twice the size of each compressed file.
5. On the file system that you just selected, create a parent directory for each product that you plan to install, for example `Tg_1`, to hold the installation directories.
6. Download all the installation files to the directories that you just created.
7. Verify that the files that you downloaded are the same size as the corresponding files on OTN.
8. Extract the files in each directory that you just created.
9. After you have extracted the required installation files, go to the "Running the Oracle Universal Installer" section on page 1-4.

Copying the Gateway Software to a Hard Disk

Before installing the gateway, you might want to copy the software to the hard disk. This enables the installation process to run faster. Before copying the installation media content to the hard disk, you must mount the installation media.

To copy the contents of the installation media to a hard disk:

1. Create a directory on your hard drive. For example:

   ```
   d:\install\Disk1
   ```

2. Copy the contents of the installation media to the directory that you just created.

3. After you have copied all the required installation files, go to the "Running the Oracle Universal Installer" section on page 1-4.

Running the Oracle Universal Installer

In most cases, you use the graphical user interface (GUI) provided by Oracle Universal Installer to install the gateway. However, you can also use Oracle Universal Installer to complete noninteractive installations, without using the GUI.

> **See Also:** Refer to Appendix A, "Using Response Files for Noninteractive Installation" for information about noninteractive installations and other advanced installation topics

Start the Installer and install the software, as follows:

1. If you are installing the software from disc, then mount the disc if it is not already mounted.

   If you are installing from a hard disk, then double-click `setup.exe` located in the directory you created for the downloaded or copied installation files.

2. When installing from the installation media, the Autorun screen automatically appears. If the Autorun screen does not appear, then:

   a. From the **Start** menu, select **Run**.

   b. Enter the following:

      ```
      DRIVE_LETTER:\autorun\autorun.exe
      ```

      In the Autorun screen, select **Install/Deinstall Products**.

3. Use the following guidelines to complete the installation:

   - Follow the instruction displayed in the Installer window. If you need additional information, click **Help**.
   - If you encounter errors while installing or linking the software, then see Appendix B, "Oracle Database Gateway Troubleshooting" for information about troubleshooting.

4. When you finish configuring, click **Exit**, then click **Yes** to exit from the Installer.

Part II

Installing and Configuring Oracle Database Gateway for Sybase

Part II, "Installing and Configuring Oracle Database Gateway for Sybase" describes how to install and configure Oracle Database Gateway for Sybase.

It contains the following chapters:

- Chapter 2, "Installing Oracle Database Gateway for Sybase"
- Chapter 3, "Configuring Oracle Database Gateway for Sybase"

2

Installing Oracle Database Gateway for Sybase

This chapter provides information about the hardware and software requirements and the installation procedure for Oracle Database Gateway for Sybase.

To install the gateway, follow these steps:

1. Ensure that the system meets all of the hardware and software requirements specified in "System Requirements for Oracle Database Gateway for Sybase" on page 2-1.
2. Run the Oracle Universal Installer.

   See "Step Through the Oracle Universal Installer" section on page 2-3 for more information about running the Oracle Universal Installer.

   Oracle Universal Installer is a menu-driven utility that guides you through the installation of the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

   See Table 2–2 for a description of the installation procedure of Oracle Database Gateway for Sybase.

System Requirements for Oracle Database Gateway for Sybase

This section provides information about the hardware and software requirements for the gateway. It contains the following sections:

Hardware Requirements

Table 2–1 lists the minimum hardware requirements for Oracle Database Gateway for Sybase.

Table 2–1 Hardware Requirements for Oracle Database Gateway for Sybase

Requirement	Value
Disk space for Oracle home	800 MB
Disk space for system drive (*SYSTEM_DRIVE*:\)	3.1 MB
Temporary disk space	125 MB

Table 2–1 (Cont.) Hardware Requirements for Oracle Database Gateway for Sybase

Requirement	Value
Physical Memory	512 MB of real memory is recommended to support the gateway. The total real memory requirement for the concurrent use of the gateway also depends on these factors: ■ The SQL statement processed by the user ■ The number of cursors currently opened against Sybase ■ The number of columns in the table being accessed
Processor	550 MHz

Checking the Hardware Requirements

To ensure that the system meets the minimum requirements, follow these steps:

1. Determine the physical RAM size. For a computer using Windows 2000, for example, open **System** in the control panel and select the **General** tab. If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.
2. Determine the size of the configured swap space (also known as paging file size). For a computer using Windows 2000, for example, open **System** in the control panel, select the **Advanced** tab, and click **Performance Options**.

   If necessary, then see your operating system documentation for information about how to configure additional swap space.
3. Determine the amount of free disk space on the system. For a computer using Windows 2000, for example, open **My Computer**, right-click the drive where the Oracle software is to be installed, and select **Properties**.
4. Determine the amount of disk space available in the `temp` directory. This is equivalent to the total amount of free disk space, minus what will be needed for the Oracle software to be installed.

   If there is less than 125 MB of disk space available in the `temp` directory, then first delete all unnecessary files. If the temp disk space is still less than 125 MB, then set the `TEMP` or `TMP` environment variable to point to a different hard drive. For a computer using Windows 2000, for example, open the **System** control panel, select the **Advanced** tab, and click **Environment Variables**.

Software Requirements

Oracle Database Gateway for Sybase is supported on the following Microsoft Windows operating systems:

- Windows 2000 with service pack 1 or higher. All editions, including Terminal Services and Windows 2000 MultiLanguage Edition (MLE), are supported.
- Windows Server 2003
- Windows Server 2003 R2
- Windows XP Professional

Windows Multilingual User Interface Pack is supported on Windows XP Professional and Windows Server 2003.

Certified Configurations

The gateway supports Sybase Adaptive Server. For the latest versions supported refer to the OTN Web site:

http://www.oracle.com/technology/products/gateways/pdf/certmatrix10g.pdf

Step Through the Oracle Universal Installer

Table 2–2 describes the installation procedure for Oracle Database Gateway for Sybase.

Table 2–2 The Oracle Universal Installer: Steps for Installing the Gateway

Screen	Response
Oracle Universal Installer: Welcome	Click **Next**.
Oracle Universal Installer: Specify Home Details	Specify a name for the installation in the Name field. You can also choose not to edit the default setting of the Name field of the Specify Home Details screen. The Path field in the Specify Home Details screen is where you specify the destination for your installation. You need not edit the path specification in the Path field. The default setting for this field points to *ORACLE_HOME*. After you set the fields in the Specify Home Details screen as necessary, click **Next** to continue. After loading the necessary information from the installation, the Oracle Universal Installer displays the Available Products screen.
Oracle Universal Installer: Available Product Components	**a.** Select **Oracle Database Gateway for Sybase 11.1.0.5.0**. **b.** Click **Next**.
Oracle Database Gateway for Sybase	**Sybase Database Server Host Name** - Specify the host name of the machine hosting the Sybase database server. **Sybase Database Server Port number** - Specify the port number of the Sybase database server **Sybase Database Name** - Specify the Sybase database name Click **Next** to continue.
Oracle Universal Installer: Summary	The Installation Summary screen enables you to review a tree list of options and components for this installation. Click **Install** to start installation.
Oracle Net Configuration Assistant: Welcome	Click **Cancel**.
Oracle Net Configuration Assistant:	Click **Yes**.
Oracle Universal Installer: Configuration Tools	Click **Exit**.
Exit	The final screen of the Oracle Universal Installer is the End of Installation screen. Click **Exit** to exit the installer.

The gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the installation log file, which is located in the `C:\Program Files\Oracle\Inventory\logs` directory.

The default file name is `InstallActions`*`YYYY-MM-DD_HH-mm-SS-AM/PM`*`.log`, where:

YYYY is year
MM is month
DD is day
HH is hour
mm is minute
SS is seconds
AM/PM is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

3

Configuring Oracle Database Gateway for Sybase

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

1. Configure the Gateway Initialization Parameter File
2. Configure Oracle Net for the Gateway
3. Configure the Oracle Database for Gateway Access
4. Create Database Links
5. Configure Two-Phase Commit
6. Create Sybase Views for Data Dictionary Support
7. Encrypt Gateway Initialization Parameter Values
8. Configure the Gateway to Access Multiple Sybase Databases

Configure the Gateway Initialization Parameter File

Perform the following tasks to configure the.gateway initialization parameter file.

1. Choose a System Identifier for the Gateway
2. Customize the Initialization Parameter File

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 Sybase database you are accessing. The SID is used as part of the file name for the initialization parameter file. The default SID is `tg4sybs`.

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

Customize the Initialization Parameter File

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

```
ORACLE_HOME\tg4sybs\admin\inittg4sybs.ora
```

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

This initialization file is for the default gateway SID. If you are not using `tg4sybs` as the gateway SID, you must rename the initialization parameter file using the SID you chose in the preceding step "Choose a System Identifier for the Gateway" on page 3-1. This default initialization parameter file is sufficient for starting the gateway, verifying a successful installation, and running the demonstration scripts.

A number of initialization parameters can be used to modify the gateway behavior. Refer to Appendix C, "Initialization Parameters" for the complete list of initialization parameters that can be set. Changes made to the initialization parameters only take effect in the next gateway session. The most important parameter is the `HS_FDS_CONNECT_INFO` which describes the connection to the non-Oracle system.

The default initialization parameter file already has an entry for this parameter. The syntax for `HS_FDS_CONNECT_INFO` is as follows:

```
HS_FDS_CONNECT_INFO=host_name:port_number/database_name
```

Where:

Variable	Description
host_name	is the host name or IP address of the machine hosting the Sybase database.
port_number	is the port number of the Sybase database server.
database_name	is the Sybase database name.

> **See Also:** Appendix C, "Initialization Parameters" and the *Oracle Heterogeneous Connectivity Administrator's Guide* for more information about customizing the initialization parameter file.

Configure Oracle Net for the Gateway

The gateway requires Oracle Net to communicate with the Oracle database. After configuring the gateway, perform the following tasks to configure Oracle Net to work with the gateway:

1. Configure Oracle Net Listener for the Gateway
2. Stop and Start the Oracle Net Listener for the Gateway

Configure Oracle Net Listener for the Gateway

The Oracle Net Listener listens for incoming requests from the Oracle database. For the Oracle Net Listener to listen for the gateway, information about the gateway must be added to the Oracle Net Listener configuration file, `listener.ora`. This file by default is located in *ORACLE_HOME*`\network\admin`, where *ORACLE_HOME* is the directory under which the gateway is installed.

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

- A list of Oracle Net addresses on which the Oracle Net Listener listens
- The executable name of the gateway that the Oracle Net Listener starts in response to incoming connection requests

Syntax of listener.ora File Entries

The Oracle database communicates with the gateway using Oracle Net and any supported protocol adapters. The following syntax of the address on which the Oracle Net Listener listens using the TCP/IP protocol adapter:

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

Where:

Variable	Description
host_name	is the name of the machine on which the gateway is installed.
port_number	specifies the port number used by the Oracle Net Listener. If you have other listeners running on the same machine, then the value of *port_number* must be different from the other listeners' port numbers.

To direct the Oracle Net Listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file.

> **Note:** You must use the same SID value in the `listener.ora` file and the `tnsnames.ora` file which will be configured in the next step.

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

Where:

Variable	Description
gateway_sid	specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.
oracle_home_directory	specifies the Oracle home directory where the gateway resides.
`tg4sybs`	specifies the executable name of the Oracle Database Gateway for Sybase.

If you already have an existing Oracle Net Listener, then add the following syntax to `SID_LIST` in the existing `listener.ora` file:

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

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

Stop and Start the Oracle Net Listener for the Gateway

You must stop and restart the Oracle Net Listener to initiate the new settings, as follows:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. If the service is already running, click **Stop** to stop it.
4. Click **Start** to start or restart the service.

Configure the Oracle Database for Gateway Access

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

To configure the Oracle database you must add connect descriptors to the `tnsnames.ora` file. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database is installed. You cannot use the Oracle Net Assistant or the Oracle Net 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.

Configuring tnsnames.ora

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is a syntax of the Oracle Net entry using the TCP/IP protocol:

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

Where:

Table 3–1 Gateway Parameters for tnsnames.ora File

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as `tg4sybs`. Check the sqlnet.ora file for the following parameter setting: `names.directory_path = (TNSNAMES)` Note: The `sqlnet.ora` file is typically stored in *ORACLE_HOME*`/network/admin`.
`TCP`	is the TCP protocol used for TCP/IP connections.
host_name	specifies the machine where the gateway is running.
port_number	matches the port number used by the Oracle Net Listener that is listening for the gateway. The Oracle Net Listener's port number can be found in the `listener.ora` file used by the Oracle Net Listener. See "Syntax of listener.ora File Entries" on page 3-3.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the Oracle Net Listener that is listening for the gateway. See "Configure Oracle Net Listener for the Gateway" on page 3-2 for more information.
`(HS=OK)`	specifies that this connect descriptor connects to a non-Oracle system.

Create Database Links

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

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 and the gateway. The connection remains established until the Oracle session ends. Another session or user can access the same database link and get a distinct connection to the gateway and Sybase database.

Database links are active for the duration of a gateway session. If you want to close a database link during a session, you can do so with the `ALTER SESSION` statement.

To access the Sybase server, you must create a database link. A public database link is the most common of database links.

```
SQL> CREATE PUBLIC DATABASE LINK dblink CONNECT TO
  2  "user" IDENTIFIED BY "password" USING 'tns_name_entry';
```

Where:

Variable	Description
dblink	is the complete database link name.
tns_name_entry	specifies the Oracle Net connect descriptor specified in the `tnsnames.ora` file that identifies the gateway

After the database link is created you can verify the connection to the Sybase database, as follows:

```
SQL> SELECT * FROM DUAL@dblink;
```

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

Configure Two-Phase Commit

The gateway supports the following transaction capabilities:

- COMMIT_CONFIRM
- READ_ONLY
- SINGLE_SITE

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

See Also: *Oracle Heterogeneous Connectivity Administrator's Guide* for information about the two-phase commit process.

To enable the COMMIT_CONFIRM transaction mode, perform the following tasks:

1. Create a Recovery Account and Password
2. Create the Transaction Log Table

The log table, called HS_TRANSACTION_LOG, is where two-phase commit transactions are recorded.

Create a Recovery Account and Password

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

Note: Oracle 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.

Target reqd in HETER sec 4.2.3 <applicable to all parts>

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

   See Also: "Customize the Initialization Parameter File" on page 3-2 for information about editing the initialization parameter file. For information about `HS_FDS_RECOVERY_ACCOUNT` and `HS_FDS_RECOVERY_PWD`, see Appendix C, "Initialization Parameters".

Create the Transaction Log Table

When configuring the gateway for two-phase commit, a table must be created in the Sybase database for logging transactions. The gateway uses the transaction log table to check the status of failed transactions that were started at the Sybase database by the gateway and registered in the table.

Note: Updates to the transaction log table cannot be part of an Oracle distributed transaction.

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

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

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

See Also: Appendix C, "Initialization Parameters" for information about the `HS_FDS_TRANSACTION_LOG` initialization parameter.

Create the transaction log table in the user account you created in "Create a Recovery Account and Password" on page 3-6. Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database where the Sybase update takes place. Also, the transaction log table must be created under the owner of the recovery account.

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

To create a transaction log table use the `tg4sybs_tx.sql` script, located in the directory *ORACLE_HOME*`\tg4sybs\admin`, where *ORACLE_HOME* is the directory under which the gateway is installed. Use isql to execute the script at the MS-DOS prompt, as follows:

```
> isql -Urecovery_account -Precovery_account_password [-Sserver] -itg4sybs_tx.sql
```

Create Sybase Views for Data Dictionary Support

To enable Oracle data dictionary translation support use the `tg4sybs_cvw.sql` script, located in the directory `$ORACLE_HOME/tg4sybs/admin` where `$ORACLE_HOME` is the directory under which the gateway is installed. You must run this script on each Sybase database that you want to access through the gateway. Use isql to execute the script, as follows:

```
$ isql -Usa_user -Psa_pwd [-Sserver] [-Ddatabase] -e -i tg4sybs_cvw.sql
```

where `sa_user` and `sa_pwd` are the Sybase system administrator user ID and password respectively.

Encrypt Gateway Initialization Parameter Values

The gateway uses user IDs and passwords to access the information in the remote database. Some user IDs 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 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.

Configure the Gateway to Access Multiple Sybase Databases

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

- The gateway is installed and configured with the default SID of `tg4sybs`.
- The gateway is configured for one Sybase database named `db1`.
- Two Sybase databases named `db2` and `db3` on a host with IP Address 204.179.79.15 are being added.

Multiple Sybase Databases Example: Configuring the Gateway

Choose One System ID for Each Sybase Database

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

- `tg4sybs2` for the gateway accessing database `db2`.

- tg4sybs3 for the gateway accessing database db3.

Create Two Initialization Parameter Files

Create an initialization parameter file for each instance of the gateway by copying the original initialization parameter file: *ORACLE_HOME*\tg4sybs\admin\inittg4sybs.ora, twice, naming one with the gateway SID for db2 and the other with the gateway SID for db3:

```
> cd ORACLE_HOME\tg4sybs\admin
> copy inittg4sybs.ora inittg4sybs2.ora
> copy inittg4sybs.ora inittg4sybs3.ora
```

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

For inittg4sybs2.ora, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15:5000/db2
```

For inittg4sybs3.ora, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15:5000/db3
```

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

Multiple Sybase Databases Example: Configuring Oracle Net Listener

Add Entries to listener.ora

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

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

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

where, oracle_home_directory is the directory where the gateway resides.

Multiple Sybase Databases Example: Stopping and Starting the Oracle Net Listener

Perform the following steps:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

Multiple Sybase Databases Example: Configuring Oracle Database for Gateway Access

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.

This example describes how to configure Oracle Net on the Oracle database for multiple gateway instances. It 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=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4sybs))
                (HS=OK))
new_db2_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4sybs2))
                 (HS=OK))
new_db3_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4sybs3))
                 (HS=OK))
```

The value for `PORT` is the TCP/IP port number of the Oracle Net Listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the Oracle Net 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 Oracle Net Listener.

Multiple Sybase Databases Example: Accessing Sybase Data

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

```
SQL> CREATE PUBLIC DATABASE LINK SYBS2 CONNECT TO
  2  "user2" IDENTIFIED BY "password2" USING 'new_db2_using';
```

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

```
SQL> CREATE PUBLIC DATABASE LINK SYBS3 CONNECT TO
  2  "user3" IDENTIFIED BY "password3" USING 'new_db3_using';
```

After the database links are created, you can verify the connection to the new Sybase databases, as in the following:

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

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

Part III

Installing and Configuring Oracle Database Gateway for Informix

Part III, "Installing and Configuring Oracle Database Gateway for Informix" describes how to install and configure Oracle Database Gateway for Informix.

It contains the following chapters:

- Chapter 4, "Installing Oracle Database Gateway for Informix"
- Chapter 5, "Configuring Oracle Database Gateway for Informix"

4

Installing Oracle Database Gateway for Informix

This chapter provides information about the hardware and software requirements and the installation procedure for Oracle Database Gateway for Informix.

To install the gateway, follow these steps:

1. Ensure that the system meets all of the hardware and software requirements specified in "System Requirements for Oracle Database Gateway for Informix" on page 4-1.
2. Run the Oracle Universal Installer.

   See "Step Through the Oracle Universal Installer" section on page 4-3 for more information about running the Oracle Universal Installer.

   Oracle Universal Installer is a menu-driven utility that guides you through the installation of the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

   See Table 4–2 for a description of the installation procedure of Oracle Database Gateway for Informix.

System Requirements for Oracle Database Gateway for Informix

This section provides information about the hardware and software requirements for the gateway. It contains the following sections:

Hardware Requirements

Table 4–1 lists the minimum hardware requirements for Oracle Database Gateway for Informix.

Table 4–1 Hardware Requirements for Oracle Database Gateway ***for Informix***

Requirement	Value
Disk space for Oracle home	800 MB
Disk space for system drive (*SYSTEM_DRIVE*:\)	3.1 MB
Temporary disk space	125 MB

Table 4–1 (Cont.) Hardware Requirements for Oracle Database Gateway *for Informix*

Requirement	Value
Physical Memory	512 MB of real memory is recommended to support the gateway. The total real memory requirement for the concurrent use of the gateway also depends on these factors: ■ The SQL statement processed by the user ■ The number of cursors currently opened against Informix ■ The number of columns in the table being accessed
Processor	550 MHz

Checking the Hardware Requirements

To ensure that the system meets the minimum requirements, follow these steps:

1. Determine the physical RAM size. For a computer using Windows 2000, for example, open **System** in the control panel and select the **General** tab. If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.
2. Determine the size of the configured swap space (also known as paging file size). For a computer using Windows 2000, for example, open **System** in the control panel, select the **Advanced** tab, and click **Performance Options**.

   If necessary, then see your operating system documentation for information about how to configure additional swap space.
3. Determine the amount of free disk space on the system. For a computer using Windows 2000, for example, open **My Computer**, right-click the drive where the Oracle software is to be installed, and select **Properties**.
4. Determine the amount of disk space available in the `temp` directory. This is equivalent to the total amount of free disk space, minus what will be needed for the Oracle software to be installed.

   If there is less than 125 MB of disk space available in the `temp` directory, then first delete all unnecessary files. If the temp disk space is still less than 125 MB, then set the `TEMP` or `TMP` environment variable to point to a different hard drive. For a computer using Windows 2000, for example, open the **System** control panel, select the **Advanced** tab, and click **Environment Variables**.

Software Requirements

Oracle Database Gateway for Informix is supported on the following Microsoft Windows operating systems:

- Windows 2000 with service pack 1 or higher. All editions, including Terminal Services and Windows 2000 MultiLanguage Edition (MLE), are supported.
- Windows Server 2003
- Windows Server 2003 R2
- Windows XP Professional

Windows Multilingual User Interface Pack is supported on Windows XP Professional and Windows Server 2003.

Certified Configurations

The gateway supports Informix Dynamic Server. For the latest versions supported refer to the OTN Web site:

http://www.oracle.com/technology/products/gateways/pdf/certmatrix10g.pdf

Step Through the Oracle Universal Installer

Table 4–2 describes the installation procedure for Oracle Database Gateway for Informix.

Table 4–2 The Oracle Universal Installer: Steps for Installing the Gateway

Screen	Response
Oracle Universal Installer: Welcome	Click **Next**.
Oracle Universal Installer: Specify Home Details	Specify a name for the installation in the Name field. You can also choose not to edit the default setting of the Name field of the Specify Home Details screen. The Path field in the Specify Home Details screen is where you specify the destination for your installation. You need not edit the path specification in the Path field. The default setting for this field points to *ORACLE_HOME*. After you set the fields in the Specify Home Details screen as necessary, click **Next** to continue. After loading the necessary information from the installation, the Oracle Universal Installer displays the Available Products screen.
Oracle Universal Installer: Available Product Components	**a.** Select **Oracle Database Gateway for Informix 11.1.0.5.0**. **b.** Click **Next**.
Oracle Database Gateway for Informix	**Informix Database Server Host Name** - Specify the host name of the machine hosting the Informix database server. **Informix Database Server Port number** - Specify the port number of the Informix database server **Informix Server Name** - Specify the Informix server name **Informix Database Name** - Specify the Informix database name Click **Next** to continue.
Oracle Universal Installer: Summary	The Installation Summary screen enables you to review a tree list of options and components for this installation. Click **Install** to start installation.
Oracle Net Configuration Assistant: Welcome	Click **Cancel**.
Oracle Net Configuration Assistant:	Click **Yes**.
Oracle Universal Installer: Configuration Tools	Click **Exit**.
Exit	The final screen of the Oracle Universal Installer is the End of Installation screen. Click **Exit** to exit the installer.

The gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the installation log file, which is located in the C:\Program Files\Oracle\Inventory\logs directory.

The default file name is `InstallActions`*`YYYY-MM-DD_HH-mm-SS-AM/PM`*`.log`, where:

YYYY is year
MM is month
DD is day
HH is hour
mm is minute
SS is seconds
AM/PM is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

5

Configuring Oracle Database Gateway for Informix

After installing the gateway, perform the following tasks to configure Oracle Database Gateway for Informix:

1. Configure the Gateway Initialization Parameter File
2. Configure Oracle Net for the Gateway
3. Configure the Oracle Database for Gateway Access
4. Create Database Links
5. Configure Two-Phase Commit
6. Encrypt Gateway Initialization Parameter Values
7. Configure the Gateway to Access Multiple Informix Databases

Configure the Gateway Initialization Parameter File

Perform the following tasks to configure the gateway initialization parameter file:

1. Choose a System Identifier for the Gateway
2. Customize the Initialization Parameter File

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 Informix database you are accessing. The SID is used as part of the file name for the initialization parameter file. The default SID is tg4ifmx.

You can define a gateway SID, but using the default of tg4ifmx is easier because you do not need to change the initialization parameter file name. However, if you want to access two Informix databases, you need two gateway SIDs, one for each instance of the gateway. If you have only one Informix database and want to access it sometimes with one set of gateway parameter settings, and other times with different gateway parameter settings, then you will need multiple gateway SIDs for the single Informix database.

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\tg4ifmx\admin\inittg4ifmx.ora

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

This initialization file is for the default gateway SID. If you are not using tg4ifmx as the gateway SID, you must rename the initialization parameter file using the SID you chose in the preceding step "Choose a System Identifier for the Gateway" on page 5-1. This default initialization parameter file is sufficient for starting the gateway, verifying a successful installation, and running the demonstration scripts.

A number of initialization parameters can be used to modify the gateway behavior. Refer to Appendix C, "Initialization Parameters" for the complete list of initialization parameters that can be set. Changes made to the initialization parameters only take effect in the next gateway session. The most important parameter is the HS_FDS_CONNECT_INFO which describes the connection to the non-Oracle system.

The default initialization parameter file already has an entry for this parameter. The syntax for HS_FDS_CONNECT_INFO is as follows:

HS_FDS_CONNECT_INFO=*host_name*:*port_number*/*server_name*/*database_name*

Where:

Variable	Description
host_name	is the host name or IP address of the machine hosting the Informix database.
port_number	is the port number of the Informix database server.
server_name	specifies the Informix database server name.
database_name	is the Informix database name.

See Also: Appendix C, "Initialization Parameters" and the *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about customizing the initialization parameter file.

Configure Oracle Net for the Gateway

The gateway requires Oracle Net to communicate with the Oracle database. After configuring the gateway, perform the following tasks to configure Oracle Net to work with the gateway:

1. Configure Oracle Net Listener for the Gateway
2. Stop and Start the Oracle Net Listener for the Gateway

Configure Oracle Net Listener for the Gateway

The Oracle Net Listener listens for incoming requests from the Oracle database. For the Oracle Net Listener to listen for the gateway, information about the gateway must be added to the Oracle Net Listener configuration file, listener.ora. This file by default is located in *ORACLE_HOME*\network\admin, where *ORACLE_HOME* is the directory under which the gateway is installed.

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

- A list of Oracle Net addresses on which the Oracle Net Listener listens

- The executable name of the gateway that the Oracle Net Listener starts in response to incoming connection requests

Syntax of listener.ora File Entries

The Oracle database communicates with the gateway using Oracle Net and any supported protocol adapters. The following syntax of the address on which the Oracle Net Listener listens using the TCP/IP protocol adapter:

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

Where:

Variable	Description
host_name	is the name of the machine on which the gateway is installed.
port_number	specifies the port number used by the Oracle Net Listener. If you have other listeners running on the same machine, then the value of *port_number* must be different from the other listeners' port numbers.

To direct the Oracle Net Listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file.

> **Note:** You must use the same SID value in the listener.ora file and the `tnsnames.ora` file which will be configured in the next step.

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (SID_NAME=gateway_sid)
         (ORACLE_HOME=oracle_home_directory)
         (PROGRAM=tg4ifmx)
      )
   )
```

Where:

Variable	Description
gateway_sid	specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.
oracle_home_directory	specifies the Oracle home directory where the gateway resides.
`tg4ifmx`	specifies the executable name of the Oracle Database Gateway for Informix.

If you already have an existing Oracle Net Listener, then add 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=tg4ifmx)
  )
)
```

See Also: *Oracle Database Net Services Administrator's Guide* for information about changing the `listener.ora` file.

Stop and Start the Oracle Net Listener for the Gateway

You must stop and restart the Oracle Net Listener to initiate the new settings, as follows:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. If the service is already running, click **Stop** to stop it.
4. Click **Start** to start or restart the service.

Configure the Oracle Database for Gateway Access

Before you use the gateway to access Informix data you must configure the Oracle database to enable communication with the gateway over Oracle Net.

To configure the Oracle database you must add connect descriptors to the `tnsnames.ora` file. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database is installed. You cannot use the Oracle Net Assistant or the Oracle Net 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.

Configuring tnsnames.ora

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is a syntax of the Oracle Net entry using the TCP/IP protocol.

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

Where:

Table 5–1 Parameters for listener.ora File

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as tg4ifmx. 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`	is the TCP protocol used for TCP/IP connections.
host_name	specifies the machine where the gateway is running.
port_number	matches the port number used by the Oracle Net Listener that is listening for the gateway. The Oracle Net Listener's port number can be found in the `listener.ora` file used by the Oracle Net Listener. See "Syntax of listener.ora File Entries" on page 5-3.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the Oracle Net Listener that is listening for the gateway. See "Configure Oracle Net Listener for the Gateway" on page 5-2 for more information.
`(HS=OK)`	specifies that this connect descriptor connects to a non-Oracle system.

Create Database Links

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

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 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 Informix 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.

To access the Informix server, you must create a database link. A public database link is the most common of database links.

```
SQL> CREATE PUBLIC DATABASE LINK dblink CONNECT TO
  2  "user" IDENTIFIED BY "password" USING 'tns_name_entry';
```

Where:

Variable	Description
dblink	is the complete database link name.
tns_name_entry	specifies the Oracle Net connect descriptor specified in the `tnsnames.ora` file that identifies the gateway

After the database link is created you can verify the connection to the Informix database, as follows:

```
SQL> SELECT * FROM DUAL@dblink;
```

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

Configure Two-Phase Commit

The gateway supports the following transaction capabilities:

- COMMIT_CONFIRM
- READ_ONLY
- SINGLE_SITE

The transaction model is set using the HS_TRANSACTION_MODEL initialization parameter. By default, the gateway runs in COMMIT_CONFIRM transaction mode. When the Informix database is updated by a transaction, the gateway becomes the commit point site. The Oracle database commits the unit of work in the Informix database after verifying that all Oracle databases in the transaction have successfully prepared the transaction. Only one gateway instance 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, perform the following tasks:

1. Create a Recovery Account and Password
2. Create the Transaction Log Table

The log table, called HS_TRANSACTION_LOG, is where two-phase commit transactions are recorded.

Create a Recovery Account and Password

For the gateway to recover distributed transactions, a recovery account and password must be set up in the Informix 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 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 Oracle Heterogeneous Connectivity Administration Guide for details.

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

   See Also: Customize the Initialization Parameter File on page 5-1 for information about editing the initialization parameter file. For information about HS_FDS_RECOVERY_ACCOUNT and HS_FDS_RECOVERY_PWD, see Appendix C, "Initialization Parameters".

Create the Transaction Log Table

When configuring the gateway for two-phase commit, a table must be created in the Informix database for logging transactions. The gateway uses the transaction log table to check the status of failed transactions that were started at the Informix database by the gateway and registered in the table.

Note: Updates to the transaction log table cannot be part of an Oracle distributed transaction.

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

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

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

See Also: Appendix C, "Initialization Parameters" for information about the HS_FDS_TRANSACTION_LOG initialization parameter.

Create the transaction log table in the user account you created in Create a Recovery Account and Password on page 5-6. Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database where the Informix 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 tg4ifmx_tx.sql script, located in the directory *ORACLE_HOME*\tg4ifmx\admin where *ORACLE_HOME* is the directory under which the gateway is installed. Use isql to execute the script at the MS-DOS prompt, as follows:

```
> isql -Urecovery_account -Precovery_account_password [-Sserver] -itg4ifmx_tx.sql
```

Encrypt Gateway Initialization Parameter Values

The gateway uses user IDs and passwords to access the information in the remote database. Some user IDs 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 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.

Configure the Gateway to Access Multiple Informix Databases

The tasks for configuring the gateway to access multiple Informix 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 `tg4ifmx`.
- The `ORACLE_HOME` environment variable is set to the directory where the gateway is installed.
- The gateway is configured for one Informix database named `db1`.
- Two Informix databases named `db2` and `db3` on a host with IP Address 204.179.79.15 are being added.

Multiple Informix Databases Example: Configuring the Gateway

Choose One System ID for Each Informix Database

A separate instance of the gateway is needed for each Informix database. Each instance needs its own gateway System ID (SID). For this example, the gateway SIDs are chosen for the instances that access the Informix databases:

- `tg4ifmx2` for the gateway accessing database `db2`.
- `tg4ifmx3` 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*`\tg4ifmx\admin\inittg4ifmx.ora`, twice, naming one with the gateway SID for `db2` and the other with the gateway SID for `db3`:

```
> cd ORACLE_HOME\tg4ifmx\admin
> copy inittg4ifmx.ora inittg4ifmx2.ora
> copy inittg4ifmx.ora inittg4ifmx3.ora
```

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

For `inittg4ifmx2.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15:3900/sr2/db2
```

For `inittg4ifmx3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15:3900/sr3/db3
```

Note: If you have multiple gateway SIDs for the same Informix 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 Informix Databases Example: Configuring Oracle Net Listener

Add Entries to listener.ora

Add two new entries to the Oracle Net 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=tg4ifmx)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4ifmx)
   )
   (SID_DESC=
      (SID_NAME=tg4ifmx2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4ifmx)
   )
   (SID_DESC=
      (SID_NAME=tg4ifmx3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4ifmx)
   )
)
```

where, `oracle_home_directory` is the directory where the gateway resides.

Multiple Informix Databases Example: Stopping and Starting the Oracle Net Listener

Perform the following steps:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

Multiple Informix Databases Example: Configuring Oracle Database for Gateway Access

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.

This example describes how to configure Oracle Net on the Oracle database for multiple gateway instances. It 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=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4ifmx))
                (HS=OK))
new_db2_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4ifmx2))
                 (HS=OK))
new_db3_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4ifmx3))
                 (HS=OK))
```

The value for `PORT` is the TCP/IP port number of the Oracle Net Listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the Oracle Net 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 Oracle Net Listener.

Multiple Informix Databases Example: Accessing Informix Data

Enter the following to create a database link for the `tg4ifmx2` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK IFMX2 CONNECT TO
  2  "user2" IDENTIFIED BY "password2" USING 'new_db2_using';
```

Enter the following to create a database link for the `tg4ifmx3` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK IFMX3 CONNECT TO
  2  "user3" IDENTIFIED BY "password3" USING 'new_db3_using';
```

After the database links are created, you can verify the connection to the new Informix databases, as in the following:

```
SQL> SELECT * FROM ALL_USERS@IFMX2;

SQL> SELECT * FROM ALL_USERS@IFMX3;
```

Part IV

Installing and Configuring Oracle Database Gateway for Teradata

Part IV, "Installing and Configuring Oracle Database Gateway for Teradata" describes how to install and configure of Oracle Database Gateway for Teradata.

It contains the following chapters:

- Chapter 6, "Installing Oracle Database Gateway for Teradata"
- Chapter 7, "Configuring Oracle Database Gateway for Teradata"

6

Installing Oracle Database Gateway for Teradata

This chapter provides information about the hardware and software requirements and the installation procedure for Oracle Database Gateway for Teradata.

To install the gateway, follow these steps:

1. Ensure that the system meets all of the hardware and software requirements specified in "System Requirements for Oracle Database Gateway for Teradata" section on page 6-1.
2. Run the Oracle Universal Installer

   See "Step Through the Oracle Universal Installer" section on page 6-3 for more information on running the Oracle Universal Installer.

   Oracle Universal Installer is a menu-driven utility that guides you through the installation of the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

   See Table 6–2 for a description of the installation procedure of Oracle Database Gateway for Teradata.

System Requirements for Oracle Database Gateway for Teradata

This chapter provides information about the hardware and software requirements for the gateway. It contains the following sections:

Hardware Requirements

Table 6–1 lists the minimum hardware requirements for Oracle Database Gateway for Teradata.

Table 6–1 Hardware Requirements for Oracle Database Gateway for Teradata

Requirement	Value
Disk space for Oracle home	800 MB
Disk space for system drive (*SYSTEM_DRIVE*:\)	3.1 MB
Temporary disk space	125 MB

Table 6–1 (Cont.) Hardware Requirements for Oracle Database Gateway for Teradata

Requirement	Value
Physical Memory	512 MB of real memory is recommended to support the gateway. The total real memory requirement for the concurrent use of the gateway also depends on these factors: ■ The SQL statement processed by the user ■ The number of cursors currently opened against Teradata ■ The number of columns in the table being accessed
Processor	550 MHz

Checking the Hardware Requirements

To ensure that the system meets the minimum requirements, follow these steps:

1. Determine the physical RAM size. For a computer using Windows 2000, for example, open **System** in the control panel and select the **General** tab. If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.
2. Determine the size of the configured swap space (also known as paging file size). For a computer using Windows 2000, for example, open **System** in the control panel, select the **Advanced** tab, and click **Performance Options**.

   If necessary, then see your operating system documentation for information about how to configure additional swap space.
3. Determine the amount of free disk space on the system. For a computer using Windows 2000, for example, open **My Computer**, right-click the drive where the Oracle software is to be installed, and select **Properties**.
4. Determine the amount of disk space available in the `temp` directory. This is equivalent to the total amount of free disk space, minus what will be needed for the Oracle software to be installed.

   If there is less than 125 MB of disk space available in the `temp` directory, then first delete all unnecessary files. If the temp disk space is still less than 125 MB, then set the `TEMP` or `TMP` environment variable to point to a different hard drive. For a computer using Windows 2000, for example, open the **System** control panel, select the **Advanced** tab, and click **Environment Variables**.

Software Requirements

Oracle Database Gateway for Teradata is supported on the following Microsoft Windows operating systems:

- Windows 2000 with service pack 1 or higher. All editions, including Terminal Services and Windows 2000 MultiLanguage Edition (MLE), are supported.
- Windows Server 2003
- Windows Server 2003 R2
- Windows XP Professional

Windows Multilingual User Interface Pack is supported on Windows XP Professional and Windows Server 2003.

Certified Configurations

Teradata client libraries are required on the machine where the gateway is installed. For the latest certified clients refer to the OTN Web site:

http://www.oracle.com/technology/products/gateways/pdf/certmatrix10g.pdf

Step Through the Oracle Universal Installer

Table 6–2 describes the installation procedure for Oracle Database Gateway for Teradata.

Table 6–2 The Oracle Universal Installer: Steps for Installing the Gateway

Screen	Response
Oracle Universal Installer: Welcome	Click **Next**.
Oracle Universal Installer: Specify Home Details	Specify a name for the installation in the Name field. You can also choose not to edit the default setting of the Name field of the Specify Home Details screen. The Path field in the Specify Home Details screen is where you specify the destination for your installation. You need not edit the path specification in the Path field. The default setting for this field points to *ORACLE_HOME*. After you set the fields in the Specify Home Details screen as necessary, click **Next** to continue. After loading the necessary information from the installation, the Oracle Universal Installer displays the Available Products screen.
Oracle Database Gateway for Teradata	**Teradata Database Server Host IP or Alias** - Specify either the host IP or alias name of the machine running the Teradata database server. **Teradata Database Server Port number** - Specify the port number of the Teradata database server **Teradata Database Name** - Specify the Teradata database name **Teradata TD_ICU_DATA Path** - Specify the local path where ICU data libraries are located (Typically `/opt/teradata/tdicu/lib` or what `$TD_ICU_DATA` is set to in `/etc/profile`). **Teradata COPLIB Path** – Specify the local path were `COPLIB` is located (Typically `/usr/lib` or what `$COPLIB` is set to in `/etc/profile`). **Teradata COPERR Path** – Specify the local path were `COPERR` is located (Typically `/usr/lib` or what `$COPERR` is set to in `/etc/profile`). Click **Next** to continue.
Oracle Universal Installer: Summary	The Installation Summary screen enables you to review a tree list of options and components for this installation. Click **Install** to start installation.
Oracle Net Configuration Assistant: Welcome	Click **Cancel**.
Oracle Net Configuration Assistant:	Click **Yes**.
Oracle Universal Installer: Configuration Tools	Click **Exit**.
Exit	The final screen of the Oracle Universal Installer is the End of Installation screen. Click **Exit** to exit the installer.

The gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the installation log file, which is located in the C:\Program Files\Oracle\Inventory\logs directory.

The default file name is `InstallActions`*`YYYY-MM-DD_HH-mm-SS-AM/PM`*`.log`, where:

YYYY is year
MM is month
DD is day
HH is hour
mm is minute
SS is seconds
AM/PM is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

7

Configuring Oracle Database Gateway for Teradata

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

1. Configure the Gateway Initialization Parameter File
2. Configure Oracle Net for the Gateway
3. Configure the Oracle Database for Gateway Access
4. Create Database Links
5. Configure Two-Phase Commit
6. Encrypt Gateway Initialization Parameter Values
7. Configure the Gateway to Access Multiple Teradata Databases

Configure the Gateway Initialization Parameter File

Perform the following tasks to configure the gateway initialization parameter file:

1. Choose a System Identifier for the Gateway
2. Customize the Initialization Parameter File

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 only one Teradata database and want to access it sometimes with one set of gateway parameter settings, and other times with different gateway parameter settings, then you will need multiple gateway SIDs for the single Teradata database.

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.

This initialization file is for the default gateway SID. If you are not using `tg4tera` as the gateway SID, you must rename the initialization parameter file using the SID you chose in the preceding Step "Choose a System Identifier for the Gateway" on page 7-1. This default initialization parameter file is sufficient for starting the gateway, verifying a successful installation, and running the demonstration scripts.

A number of initialization parameters can be used to modify the gateway behavior. Refer to Appendix C, "Initialization Parameters" for the complete list of initialization parameters that can be set. Changes made to the initialization parameters only take effect in the next gateway session. The most important parameter is the `HS_FDS_CONNECT_INFO` which describes the connection to the non-Oracle system.

The default initialization parameter file already has an entry for this parameter. The syntax for `HS_FDS_CONNECT_INFO` is as follows:

`HS_FDS_CONNECT_INFO=`*`ip_address`*`:`*`port_number`*`[/`*`database_name`*`]`

Where:

Variable	Description
ip_address	is the IP address of the machine hosting the Teradata database.
port_number	is the port number of the Teradata database server.
database_name	is the Teradata database name. The *database_name* variable is optional

See Also: Appendix C, "Initialization Parameters" and the *Oracle Heterogeneous Connectivity Administrator's Guide* for more information about customizing the initialization parameter file.

Configure Oracle Net for the Gateway

The gateway requires Oracle Net to communicate with the Oracle database. After configuring the gateway, perform the following tasks to configure Oracle Net to work with the gateway:

1. Configure Oracle Net Listener for the Gateway
2. Stop and Start the Oracle Net Listener for the Gateway

Configure Oracle Net Listener for the Gateway

The Oracle Net Listener listens for incoming requests from the Oracle database. For the Oracle Net Listener to listen for the gateway, information about the gateway must be added to the Oracle Net Listener configuration file, `listener.ora`. This file by default is located in *ORACLE_HOME*\network\admin, where *ORACLE_HOME* is the directory under which the gateway is installed.

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

- A list of Oracle Net addresses on which the Oracle Net Listener listens
- The executable name of the gateway that the Oracle Net Listener starts in response to incoming connection requests

Syntax of listener.ora File Entries

The Oracle database communicates with the gateway using Oracle Net and any supported protocol adapters. The following syntax of the address on which the Oracle Net Listener listens using the TCP/IP protocol adapter:

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

Where:

Variable	Description
host_name	is the name of the machine on which the gateway is installed.
port_number	specifies the port number used by the Oracle Net Listener. If you have other listeners running on the same machine, then the value of *port_number* must be different from the other listeners' port numbers.

To direct the Oracle Net Listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file.

> **Note:** You must use the same SID value in the `listener.ora` file and the `tnsnames.ora` file which will be configured in the next step.

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (SID_NAME=gateway_sid)
         (ORACLE_HOME=oracle_home_directory)
         (PROGRAM=tg4tera)
      )
   )
```

Where:

Variable	Description
gateway_sid	specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.
oracle_home_directory	specifies the Oracle home directory where the gateway resides.
`tg4tera`	specifies the Oracle Database Gateway for Teradata.

If you already have an existing Oracle Net Listener, then add 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)
   )
)
```

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

Stop and Start the Oracle Net Listener for the Gateway

You must stop and restart the Oracle Net Listener to initiate the new settings, as follows:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. If the service is already running, click **Stop** to stop it.
4. Click **Start** to start or restart the service.

Configure the Oracle Database for Gateway Access

Before you use the gateway to access Teradata data you must configure the Oracle database to enable communication with the gateway over Oracle Net.

To configure the Oracle database you must add connect descriptors to the `tnsnames.ora` file. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database is installed. You cannot use the Oracle Net Assistant or the Oracle Net 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.

Configuring tnsnames.ora

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is a syntax of the Oracle Net entry using the TCP/IP protocol:

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

Where:

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as `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	is the TCP protocol used for TCP/IP connections.
host_name	specifies the machine where the gateway is running.
port_number	matches the port number used by the Oracle Net Listener that is listening for the gateway. The Oracle Net Listener's port number can be found in the `listener.ora` file used by the Oracle Net Listener. See "Syntax of listener.ora File Entries" on page 7-3.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the Oracle Net Listener that is listening for the gateway. See "Configure Oracle Net Listener for the Gateway" on page 7-2 for more information.
`(HS=OK)`	specifies that this connect descriptor connects to a non-Oracle system.

Create Database Links

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

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 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.

To access the Teradata server, you must create a database link. A public database link is the most common of database links.

```
SQL> CREATE PUBLIC DATABASE LINK dblink CONNECT TO
  2  "user" IDENTIFIED BY "password" USING 'tns_name_entry';
```

where, *dblink* is the complete database link name and *tns_name_entry* specifies the Oracle Net connect descriptor specified in the tnsnames.ora file that identifies the gateway .

After the database link is created you can verify the connection to the Teradata database, as follows:

```
SQL> SELECT * FROM DUAL@dblink;
```

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

Configure Two-Phase Commit

The gateway supports the following transaction capabilities:

- COMMIT_CONFIRM
- READ_ONLY
- SINGLE_SITE

The transaction model is set using the HS_TRANSACTION_MODEL initialization parameter. 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 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 instance can participate in an Oracle two-phase commit transaction as the commit point site.

> **See Also:** *Oracle Heterogeneous Connectivity Administrator's Guide* for information about the two-phase commit process.

To enable the COMMIT_CONFIRM transaction mode, perform the following tasks:

1. Create a Recovery Account and Password
2. Create the Transaction Log Table

The log table, called HS_TRANSACTION_LOG, is where two-phase commit transactions are recorded.

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 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: "Customize the Initialization Parameter File" on page 7-1 for information about editing the initialization parameter file. For information about HS_FDS_RECOVERY_ACCOUNT and HS_FDS_RECOVERY_PWD, see Appendix C, "Initialization Parameters".

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.

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 C, "Initialization Parameters" for information about the HS_FDS_TRANSACTION_LOG initialization parameter.

Create the transaction log table in the user account you created in "Create a Recovery Account and Password" on page 7-6. 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.

Encrypt Gateway Initialization Parameter Values

The gateway uses user IDs and passwords to access the information in the remote database. Some user IDs 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 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.

Configure the Gateway to Access 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 gateway is configured for one Teradata database named `db1`.
- Two Teradata databases named `db2` and `db3` on a host with IP Address 204.179.79.15 are being added.

Multiple Teradata Databases Example: Configuring the Gateway

Choose One System ID for Each Teradata Database

A separate instance of the gateway is needed for each Teradata database. 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
> copy inittg4tera.ora inittg4tera2.ora
> copy 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=204.179.79.15:1025/db2
```

For `inittg4tera3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15:1025/db3
```

> **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 Teradata Databases Example: Configuring Oracle Net Listener

Add Entries to listener.ora

Add two new entries to the Oracle Net 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)
   )
   (SID_DESC=
      (SID_NAME=tg4tera2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4tera)
   )
   (SID_DESC=
      (SID_NAME=tg4tera3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4tera)
   )
)
```

Multiple Teardata Databases Example: Stopping and Starting the Oracle Net Listener

Perform the following steps:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

Multiple Teradata Databases Example: Configuring Oracle Database for Gateway Access

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.

This example describes how to configure Oracle Net on the Oracle database for multiple gateway instances. It 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=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4tera))
                (HS=OK))
new_db2_using=(DESCRIPTION=
```

```
                (ADDRESS=
                  (PROTOCOL=TCP)
                  (PORT=port_number)
                  (HOST=host_name))
                  (CONNECT_DATA=
                      (SID=tg4tera2))
                  (HS=OK))
new_db3_using=(DESCRIPTION=
                (ADDRESS=
                  (PROTOCOL=TCP)
                  (PORT=port_number)
                  (HOST=host_name))
                  (CONNECT_DATA=
                      (SID=tg4tera3))
                  (HS=OK))
```

The value for `PORT` is the TCP/IP port number of the Oracle Net Listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the Oracle Net 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 Oracle Net Listener.

Multiple Teradata 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';
```

After the database links are created, you can verify the connection to the new Teradata databases, as in the following:

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

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

Part V

Installing and Configuring Oracle Database Gateway for Microsoft SQL Server

Part V, "Installing and Configuring Oracle Database Gateway for Microsoft SQL Server" describes how to install and configure of Oracle Database Gateway for Microsoft SQL Server

It contains the following chapters:

- Chapter 8, "Installing Oracle Database Gateway for Microsoft SQL Server"
- Chapter 9, "Configuring Oracle Database Gateway for Microsoft SQL Server"

8

Installing Oracle Database Gateway for Microsoft SQL Server

This chapter provides information about the hardware and software requirements and the installation procedure for Oracle Database Gateway for Microsoft SQL Server.

To install the Oracle Database Gateway for Microsoft SQL Server, follow these steps:

1. Ensure that the system meets all of the hardware and software requirements specified in "System Requirements for Oracle Database Gateway for Microsoft SQL Server" section on page 8-1.
2. Run the Oracle Universal Installer.

   See "Step Through the Oracle Universal Installer" section on page 8-3 for more information on running the Oracle Universal Installer.

   Oracle Universal Installer is a menu-driven utility that guides you through the installation of the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

   See Table 8–2 for description of the installation procedure of Oracle Database Gateway for Microsoft SQL Server.

System Requirements for Oracle Database Gateway for Microsoft SQL Server

This section provides information about the hardware and software requirements for the gateway. It contains the following sections:

Hardware Requirements

Table 8–1 lists the minimum hardware requirements for Oracle Database Gateway for Microsoft SQL Server.

Table 8–1 Hardware Requirements for Oracle Database Gateway ***for Microsoft SQL Server***

Requirement	Value
Disk space for Oracle home	800 MB

Table 8–1 (Cont.) Hardware Requirements for Oracle Database Gateway ***for Microsoft SQL Server***

Requirement	Value
Disk space for system drive (*SYSTEM_DRIVE*:\)	3.1 MB
Temporary disk space	125 MB
Physical Memory	512 MB of real memory is recommended to support the gateway. The total real memory requirement for the concurrent use of the gateway also depends on these factors: ■ The SQL statement processed by the user ■ The number of cursors currently opened against SQL Server ■ The number of columns in the table being accessed
Processor	550 MHz

Checking the Hardware Requirements

To ensure that the system meets the minimum requirements, follow these steps:

1. Determine the physical RAM size. For a computer using Windows 2000, for example, open **System** in the control panel and select the **General** tab. If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.
2. Determine the size of the configured swap space (also known as paging file size). For a computer using Windows 2000, for example, open **System** in the control panel, select the **Advanced** tab, and click **Performance Options**.

   If necessary, then see your operating system documentation for information about how to configure additional swap space.
3. Determine the amount of free disk space on the system. For a computer using Windows 2000, for example, open **My Computer**, right-click the drive where the Oracle software is to be installed, and select **Properties**.
4. Determine the amount of disk space available in the `temp` directory. This is equivalent to the total amount of free disk space, minus what will be needed for the Oracle software to be installed.

   If there is less than 125 MB of disk space available in the `temp` directory, then first delete all unnecessary files. If the temp disk space is still less than 125 MB, then set the `TEMP` or `TMP` environment variable to point to a different hard drive. For a computer using Windows 2000, for example, open the **System** control panel, select the **Advanced** tab, and click **Environment Variables**.

Software Requirements

Oracle Database Gateway for Microsoft SQL Server is supported on the following Microsoft Windows operating systems:

- Windows 2000 with service pack 1 or higher. All editions, including Terminal Services and Windows 2000 MultiLanguage Edition (MLE), are supported.
- Windows Server 2003
- Windows Server 2003 R2
- Windows XP Professional

Windows Multilingual User Interface Pack is supported on Windows XP Professional and Windows Server 2003.

Certified Configurations

The gateway supports Microsoft SQL Server. For the latest versions supported refer to the OTN Web site:

http://www.oracle.com/technology/products/gateways/pdf/certmatrix10g.pdf

Step Through the Oracle Universal Installer

Step Through the Oracle Universal Installer

Table 8–2 describes the installation procedure for Oracle Database Gateway for Microsoft SQL Server

Table 8–2 The Oracle Universal Installer: Steps for Installing the Gateway

Screen	Response
Oracle Universal Installer: Welcome	Click **Next**.
Oracle Universal Installer: Specify Home Details	Specify a name for the installation in the Name field. You can also choose not to edit the default setting of the Name field of the Specify Home Details screen. The Path field in the Specify Home Details screen is where you specify the destination for your installation. You need not edit the path specification in the Path field. The default setting for this field points to *ORACLE_HOME*. After you set the fields in the Specify Home Details screen as necessary, click **Next** to continue. After loading the necessary information from the installation, the Oracle Universal Installer displays the Available Products screen.
Oracle Database Gateway for Microsoft SQL Server	**SQL Server Database Server Host Name** - Specify the host name of the machine hosting the Microsoft SQL Server database. **SQL Server Database Server Port number** - Specify the port number of the SQL Server database server **SQL Server Database Name** - Specify the SQL Server database name Click **Next** to continue.
Oracle Universal Installer: Summary	The Installation Summary screen enables you to review a tree list of options and components for this installation. Click **Install** to start installation.
Oracle Net Configuration Assistant: Welcome	Click **Cancel**.
Oracle Net Configuration Assistant:	Click **Yes**.
Oracle Universal Installer: Configuration Tools	Click **Exit**.
Exit	The final screen of the Oracle Universal Installer is the End of Installation screen. Click **Exit** to exit the installer.

The gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the

installation log file, which is located in the C:\Program Files\Oracle\Inventory\logs directory.

The default file name is `InstallActions`*`YYYY-MM-DD_HH-mm-SS-AM/PM`*`.log`, where:

`YYYY` is year
`MM` is month
`DD` is day
`HH` is hour
`mm` is minute
`SS` is seconds
`AM/PM` is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

9

Configuring Oracle Database Gateway for Microsoft SQL Server

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

1. Configure the Gateway Initialization Parameter File
2. Configure Oracle Net for the Gateway
3. Configure the Oracle Database for Gateway Access
4. Create Database Links
5. Configure Two-Phase Commit
6. Create SQL Server Views for Data Dictionary Support
7. Encrypt Gateway Initialization Parameter Values
8. Configure the Gateway to Access Multiple SQL Server Databases

Configure the Gateway Initialization Parameter File

Perform the following tasks to configure the gateway initialization parameter file:

1. Choose a System Identifier for the Gateway
2. Customize the Initialization Parameter File

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 SQL Server database you are accessing. The SID is used as part of the file name for the initialization parameter file. The default SID is tg4msql.

You can define a gateway SID, but using the default of tg4msql is easier because you do not need to change the initialization parameter file name. However, if you want to access two SQL Server databases, you need two gateway SIDs, one for each instance of the gateway. If you have only one SQL Server database and want to access it sometimes with one set of gateway parameter settings, and other times with different gateway parameter settings, then you will need multiple gateway SIDs for the single SQL Server database.

Customize the Initialization Parameter File

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

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

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

This initialization file is for the default gateway SID. If you are not using `tg4msql` as the gateway SID, you must rename the initialization parameter file using the SID you chose in the preceding step "Choose a System Identifier for the Gateway" on page 9-1. This default initialization parameter file is sufficient for starting the gateway, verifying a successful installation, and running the demonstration scripts.

A number of initialization parameters can be used to modify the gateway behavior. Refer to Appendix C, "Initialization Parameters" for the complete list of initialization parameters that can be set. Changes made to the initialization parameters only take effect in the next gateway session. The most important parameter is the `HS_FDS_CONNECT_INFO` which describes the connection to the non-Oracle system.

The default initialization parameter file already has an entry for this parameter. The syntax for `HS_FDS_CONNECT_INFO` is as follows:

```
HS_FDS_CONNECT_INFO= host_name/[instance_name][/database_name]
```

Where:

Variable	Description
host_name	is the host name or IP address of the machine hosting the SQL Server database.
instance_name	is the instance of SQL Server running on the machine.
database_name	is the SQL Server Database database name.

Both *instance_name* and *database_name* are optional. If *instance_name* is omitted and *database_name* is provided, the slash (/) is required. This can be shown as follows:

```
HS_FDS_CONNECT_INFO= host_name//database_name
```

> **See Also:** Appendix C, "Initialization Parameters" and *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about customizing the initialization parameter file.

Configure Oracle Net for the Gateway

The gateway requires Oracle Net to communicate with the Oracle database. After configuring the gateway, perform the following tasks to configure Oracle Net to work with the gateway:

1. Configure Oracle Net Listener for the Gateway
2. Stop and Start the Oracle Net Listener for the Gateway

Configure Oracle Net Listener for the Gateway

The Oracle Net Listener listens for incoming requests from the Oracle database. For the Oracle Net Listener to listen for the gateway, information about the gateway must be added to the Oracle Net Listener configuration file, `listener.ora`. This file by default is located in *ORACLE_HOME*`\network\admin`, where *ORACLE_HOME* is the directory under which the gateway is installed.

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

- A list of Oracle Net addresses on which the Oracle Net Listener listens
- The executable name of the gateway that the Oracle Net Listener starts in response to incoming connection requests

Syntax of listener.ora File Entries

The Oracle database communicates with the gateway using Oracle Net and any supported protocol adapters. The following syntax of the address on which the Oracle Net Listener listens using the TCP/IP protocol adapter:

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

Where:

Variable	Description
host_name	specifies the name of the machine on which the gateway is installed.
port_number	specifies the port number used by the Oracle Net Listener. If you have other listeners running on the same machine, then the value of *port_number* must be different from the other listeners' port numbers.

To direct the Oracle Net Listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file.

> **Note:** You must use the same SID value in the `listener.ora` file and the `tnsnames.ora` file which will be configured in the next step.

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (SID_NAME=gateway_sid)
         (ORACLE_HOME=oracle_home_directory)
         (PROGRAM=tg4msql)
      )
   )
```

Where:

Variable	Description
gateway_sid	specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.
oracle_home_directory	specifies the Oracle home directory where the gateway resides.
`tg4msql`	specifies the executable name of the Oracle Database Gateway for Microsoft SQL Server.

If you already have an existing Oracle Net Listener, then add the following syntax to `SID_LIST` in the existing `listener.ora` file:

```
SID_LIST_LISTENER=
(SID_LIST=
   (SID_DESC=.
     .
   )
   (SID_DESC=.
     .
   )
   (SID_DESC=
      (SID_NAME=gateway_sid)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
)
```

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

Stop and Start the Oracle Net Listener for the Gateway

You must stop and restart the Oracle Net Listener to initiate the new settings, as follows:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. If the service is already running, click **Stop** to stop it.
4. Click **Start** to start or restart the service.

Configure the Oracle Database for Gateway Access

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

To configure the Oracle database you must add connect descriptors to the `tnsnames.ora` file. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database is installed. You cannot use the Oracle Net Assistant or the Oracle Net 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.

Configuring tnsnames.ora

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is a syntax of the Oracle Net entry using the TCP/IP protocol:

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

Where:

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as tg4msql. Check the `sqlnet.ora` file in the Oracle database's `ORACLE_HOME` for the following lines: ▪ `names.directory_path = (TNSNAMES, HOSTNAME)` ▪ `names.default_domain = world` ▪ `name.default_zone = world` **Note**: If the Oracle database is on Microsoft Windows, the file is *ORACLE_HOME*`\network\admin\sqlnet.ora`. If the `sqlnet.ora` file has these lines, *connect_descriptor* must end with the extension `.world`.
`TCP`	is the TCP protocol used for TCP/IP connections.
host_name	specifies the machine where the gateway is running.
port_number	matches the port number used by the Oracle Net Listener that is listening for the gateway. The Oracle Net Listener's port number can be found in the `listener.ora` file used by the Oracle Net Listener. See "Syntax of listener.ora File Entries" on page 9-3.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the Oracle Net Listener that is listening for the gateway. See"Configure Oracle Net Listener for the Gateway" on page 9-3 for more information.
`(HS=OK)`	specifies that this connect descriptor connects to a non-Oracle system.

Create Database Links

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

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 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 SQL Server database.

Database links are active for the duration of a gateway session. If you want to close a database link during a session, you can do so with the `ALTER SESSION` statement.

To access the SQL Server, you must create a database link. A public database link is the most common of database links.

```
SQL> CREATE PUBLIC DATABASE LINK dblink CONNECT TO
  2  "user" IDENTIFIED BY "password" USING 'tns_name_entry';
```

Where:

Variable	Description
dblink	is the complete database link name.
tns_name_entry	specifies the Oracle Net connect descriptor specified in the `tnsnames.ora` file that identifies the gateway

After the database link is created you can verify the connection to the SQL Server database, as follows:

```
SQL> SELECT * FROM DUAL@dblink;
```

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

Configure Two-Phase Commit

The gateway supports the following transaction capabilities:

- `COMMIT_CONFIRM`
- `READ_ONLY`
- `SINGLE_SITE`

The transaction model is set using the HS_TRANSACTION_MODEL initialization parameter. By default, the gateway runs in `COMMIT_CONFIRM` transaction mode. When the SQL Server database is updated by a transaction, the gateway becomes the commit point site. The Oracle database commits the unit of work in the SQL Server database after verifying that all Oracle databases in the transaction have successfully prepared the transaction. Only one gateway instance 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, perform the following tasks:

1. Create a Recovery Account and Password
2. Create the Transaction Log Table

The log table, called `HS_TRANSACTION_LOG`, is where two-phase commit transactions are recorded.

Create a Recovery Account and Password

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

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

See Also: "Customize the Initialization Parameter File" on page 9-2 for information about editing the initialization parameter file. For information about HS_FDS_RECOVERY_ACCOUNT and HS_FDS_RECOVERY_PWD, see Appendix C, "Initialization Parameters".

Create the Transaction Log Table

When configuring the gateway for two-phase commit, a table must be created in the SQL Server database for logging transactions. The gateway uses the transaction log table to check the status of failed transactions that were started at the SQL Server database by the gateway and registered in the table.

Note: Updates to the transaction log table cannot be part of an Oracle distributed transaction.

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

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

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

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

Create the transaction log table in the user account you created in "Create a Recovery Account and Password" on page 9-7. Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database where the SQL Server update takes place. Also, the transaction log table must be created under the owner of the recovery account.

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

To create a transaction log table use the tg4msql_tx.sql script, located in the directory *ORACLE_HOME*\tg4msql\admin where *ORACLE_HOME* is the directory under which the gateway is installed. Use isql to execute the script at the MS-DOS prompt, as follows:

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

Create SQL Server Views for Data Dictionary Support

To enable Oracle data dictionary translation support use the tg4msql_cvw.sql script, located in the directory $ORACLE_HOME/tg4msql/admin where $ORACLE_HOME is the directory under which the gateway is installed. You must run this script on each SQL Server database that you want to access through the gateway. Use isql to execute the script, as follows:

```
$ isql -Usa_user -Psa_pwd [-Sserver] [-ddatabase] -e -i tg4msql_cvw.sql
```

where sa_user and sa_pwd are the SQL Server system administrator user ID and password respectively.

Encrypt Gateway Initialization Parameter Values

The gateway uses user IDs and passwords to access the information in the remote database. Some user IDs 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 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.

Configure the Gateway to Access Multiple SQL Server Databases

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

- The gateway is installed and configured with the default SID of `tg4msql`
- The gateway is configured for one SQL Server database named `db1`
- Two SQL Server databases named `db2` and `db3` on a host with IP Address 204.179.79.15 are being added

Multiple SQL Server Databases Example: Configuring the Gateway

Choose One System ID for Each SQL Server Database

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

- `tg4msql2` for the gateway accessing database `db2`
- `tg4msql3` for the gateway accessing database `db3`

Create Two Initialization Parameter Files

Create an initialization parameter file for each instance of the gateway by copying the original initialization parameter file: *ORACLE_HOME*`\tg4msql\admin\inittg4msql.ora`, twice, naming one with the gateway SID for `db2` and the other with the gateway SID for `db3`:

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

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

For `inittg4msql2.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15//db2
```

For `inittg4msql3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=204.179.79.15//db3
```

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

Multiple SQL Server Databases Example: Configuring Oracle Net Listener

Add Entries to listener.ora

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

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

```
SID_LIST_LISTENER=
(SID_LIST=
   (SID_DESC=
      (SID_NAME=tg4msql)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
   (SID_DESC=
      (SID_NAME=tg4msql2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
   (SID_DESC=
      (SID_NAME=tg4msql3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4msql)
   )
)
```

where, `oracle_home_directory` is the directory where the gateway resides.

Multiple SQL Server Databases Example: Stopping and Starting the Oracle Net Listener

Perform the following steps:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

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

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.

This example describes how to configure Oracle Net on the Oracle database for multiple gateway instances. It 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=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4msql))
                (HS=OK))
new_db2_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
```

```
                    (SID=tg4msql2))
                  (HS=OK))
new_db3_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                    (SID=tg4msql3))
                 (HS=OK))
```

The value for `PORT` is the TCP/IP port number of the Oracle Net Listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the Oracle Net 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 Oracle Net Listener.

Multiple SQL Server Databases Example: Accessing SQL Server Data

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

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

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

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

After the database links are created, you can verify the connection to the new SQL Server databases, as in the following:

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

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

Part VI

Installing and Configuring Oracle Database Gateway for ODBC

Part VI, "Installing and Configuring Oracle Database Gateway for ODBC" describes how to install and configure Oracle Database Gateway for ODBC on Microsoft Windows.

It contains the following chapters:

- Chapter 10, "Installing Oracle Database Gateway for ODBC"
- Chapter 11, "Configuring Oracle Database Gateway for ODBC"

10

Installing Oracle Database Gateway for ODBC

This chapter provides information about the hardware and software requirements and the installation procedure for Oracle Database Gateway for ODBC.

To install Oracle Database Gateway for ODBC, follow these steps:

1. Ensure that the system meets all of the hardware and software requirements specified in "System Requirements for Oracle Database Gateway for ODBC" on page 10-1.
2. Run the Oracle Universal Installer.

   See "Step Through the Oracle Universal Installer" section on page 10-3 for more information about running the Oracle Universal Installer.

   Oracle Universal Installer is a menu-driven utility that guides you through the installation of Oracle Database Gateway for ODBC by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

   See Table 10–2 for a description of the installation procedure of Oracle Database Gateway for ODBC.

System Requirements for Oracle Database Gateway for ODBC

This section provides information about the hardware and software requirements for Oracle Database Gateway for ODBC. It contains the following sections:

Hardware Requirements

Table 10–1 lists the minimum hardware requirements for Oracle Database Gateway for ODBC.

Table 10–1 Hardware Requirements for Oracle Database Gateway for ODBC

Requirement	Value
Disk space for Oracle home	800 MB
Disk space for system drive (*SYSTEM_DRIVE*:\)	3.1 MB
Temporary disk space	125 MB

Table 10–1 (Cont.) Hardware Requirements for Oracle Database Gateway for ODBC

Requirement	Value
Physical Memory	512 MB of real memory is recommended to support Oracle Database Gateway for ODBC. The total real memory requirement for the concurrent use of the gateway also depends on these factors: ■ The SQL statement processed by the user ■ The number of cursors currently opened against the non-Oracle system ■ The number of columns in the table being accessed
Processor	550 MHz

Checking the Hardware Requirements

To ensure that the system meets the minimum requirements, follow these steps:

1. Determine the physical RAM size. For a computer using Windows 2000, for example, open **System** in the control panel and select the **General** tab. If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.
2. Determine the size of the configured swap space (also known as paging file size). For a computer using Windows 2000, for example, open **System** in the control panel, select the **Advanced** tab, and click **Performance Options**.

   If necessary, then see your operating system documentation for information about how to configure additional swap space.
3. Determine the amount of free disk space on the system. For a computer using Windows 2000, for example, open **My Computer**, right-click the drive where the Oracle software is to be installed, and select **Properties**.
4. Determine the amount of disk space available in the `temp` directory. This is equivalent to the total amount of free disk space, minus what will be needed for the Oracle software to be installed.

   If there is less than 125 MB of disk space available in the `temp` directory, then first delete all unnecessary files. If the temp disk space is still less than 125 MB, then set the `TEMP` or `TMP` environment variable to point to a different hard drive. For a computer using Windows 2000, for example, open the **System** control panel, select the **Advanced** tab, and click **Environment Variables**.

Software Requirements

Oracle Database Gateway for ODBC is supported on the following Microsoft Windows operating systems:

- Windows 2000 with service pack 1 or higher. All editions, including Terminal Services and Windows 2000 MultiLanguage Edition (MLE), are supported.
- Windows Server 2003
- Windows Server 2003 R2
- Windows XP Professional

Windows Multilingual User Interface Pack is supported on Windows XP Professional and Windows Server 2003.

Certified Configurations

For the latest certified configuration refer to the OTN Web site:

http://www.oracle.com/technology/products/gateways/pdf/certmatrix10g.pdf

Step Through the Oracle Universal Installer

Table 10–2 describes the installation procedure for Oracle Database Gateway for ODBC.

Table 10–2 The Oracle Universal Installer: Steps for Installing Oracle Database Gateway for ODBC

Screen	Response
Oracle Universal Installer: Welcome	Click **Next**.
Oracle Universal Installer: Specify Home Details	Specify a name for the installation in the Name field. You can also choose not to edit the default setting of the Name field of the Specify Home Details screen. The Path field in the Specify Home Details screen is where you specify the destination for your installation. You need not edit the path specification in the Path field. The default setting for this field points to *ORACLE_HOME*. After you set the fields in the Specify Home Details screen as necessary, click **Next** to continue. After loading the necessary information from the installation, the Oracle Universal Installer displays the Available Products screen.
Oracle Universal Installer: Available Product Components	**a.** Select **Oracle Database Gateway for ODBC 11.1.0.5.0**. **b.** Click **Next**.
Oracle Universal Installer: Summary	The Installation Summary screen enables you to review a tree list of options and components for this installation. Click **Install** to start installation.
Oracle Net Configuration Assistant: Welcome	Click **Cancel**.
Oracle Net Configuration Assistant:	Click **Yes**.
Oracle Universal Installer: Configuration Tools	Click **Exit**.
Exit	The final screen of the Oracle Universal Installer is the End of Installation screen. Click **Exit** to exit the installer.

The gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the installation log file, which is located in the `C:\Program Files\Oracle\Inventory\logs` directory.

The default file name is `InstallActions`*`YYYY-MM-DD_HH-mm-SS-AM/PM`*`.log`, where:

YYYY is year
MM is month
DD is day
HH is hour
mm is minute
SS is seconds

AM/PM is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

11

Configuring Oracle Database Gateway for ODBC

After installing the gateway and the ODBC driver for the non-Oracle system, perform the following tasks to configure Oracle Database Gateway for ODBC:

1. Configure the Gateway Initialization Parameter File
2. Configure Oracle Net for the Gateway
3. Configure the Oracle Database for Gateway Access
4. Create Database Links
5. Encrypt Gateway Initialization Parameter Values
6. Configure the Gateway to Access Multiple ODBC Data Sources

Configure the Gateway Initialization Parameter File

Perform the following tasks to configure the gateway initialization file:

1. Create the Initialization Parameter File
2. Set the Initialization Parameter Values

Create the Initialization Parameter File

You must create an initialization file for your Oracle Database Gateway for ODBC . Oracle supplies a sample initialization file, `inithsodbc.ora`. The sample file is stored in the *`ORACLE_HOME`*`\hs\admin` directory.

To create an initialization file for the ODBC gateway, copy the sample initialization file and rename it to `init`*`sid`*`.ora`, where *`sid`* is the system identifier(SID) you want to use for the instance of the non-Oracle system to which the gateway connects.

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 ODBC source you are accessing.

If you want to access two ODBC sources, you need two gateway SIDs, one for each instance of the gateway. If you have only one ODBC source but want to access it sometimes with one set of gateway parameter settings, and other times with different gateway parameter settings, then you will need multiple gateway SIDs for the single ODBC source. The SID is used as part of the file name for the initialization parameter file.

Set the Initialization Parameter Values

After the initialization file has been created, you must set the initialization parameter values. A number of initialization parameters can be used to modify the gateway behavior. You must set the HS_FDS_CONNECT_INFO initialization parameter. Other initialization parameters have defaults or are optional. You can use the default values and omit the optional parameters, or you can specify the parameters with values tailored for your installation. Refer to Appendix C, "Initialization Parameters" for the complete list of initialization parameters that can be set. Changes made to the initialization parameters only take effect in the next gateway session.

The HS_FDS_CONNECT_INFO initialization parameter specifies the information required for connecting to the non-Oracle system. Set the HS_FDS_CONNECT_INFO as follows:

```
HS_FDS_CONNECT_INFO=dsn_value
```

where *dsn_value* is the name of the system DSN defined in the Windows ODBC Data Source Administrator.

> **Note:** Before deciding whether to accept the default values or to change them, see Appendix C, "Initialization Parameters" for detailed information about all the initialization parameters.

Example: Setting Initialization Parameter Values

Assume that a system DSN has been defined in the Windows ODBC Data Source Administrator. In order to connect to this SQL Server database through the gateway, the following line is required in the init*sid*.ora file:

```
HS_FDS_CONNECT_INFO=sqlserver7
```

sqlserver7 is the name of the system DSN defined in the Windows ODBC Data Source Administrator.

The following procedure enables you to define a system DSN in the Windows ODBC Data Source Administrator, version 3.5:

1. From the **Start** menu, choose **Settings** > **Control Panel** and select **ODBC** .
2. Select the **System DSN** tab page to display the system data sources.
3. Click **Add**.
4. From the list of installed ODBC drivers, select the name of the driver that the data source will use. For example, select **SQL Server**.
5. Click **Finish**.
6. Enter a name for the DSN and an optional description. Enter other information depending on the ODBC driver. For example, for SQL Server enter the SQL Server machine name.

   > **Note:** The name entered for the DSN must match the value of the initialization parameter HS_FDS_CONNECT_INFO that is specified in init*sid*.ora.

7. Refer to your ODBC driver documentation and follow the prompts to complete configuration of the DSN.

8. After creating the system DSN, click **OK** to exit the ODBC Data Source Administrator.

Note: If the ODBC driver supports Quoted Identifiers or Delimited Identifiers it should be turned on.

Configure Oracle Net for the Gateway

The gateway requires Oracle Net to communicate with the Oracle database. After configuring the gateway, perform the following tasks to configure Oracle Net to work with the gateway:

1. Configure Oracle Net Listener for the Gateway
2. Stop and Start the Oracle Net Listener for the Gateway

Configure Oracle Net Listener for the Gateway

The Oracle Net Listener listens for incoming requests from the Oracle database. For the Oracle Net Listener to listen for the gateway, information about the gateway must be added to the Oracle Net Listener configuration file, `listener.ora`. This file by default is located in *ORACLE_HOME*`\network\admin`, where *ORACLE_HOME* is the directory under which the gateway is installed.

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

- A list of Oracle Net addresses on which the Oracle Net Listener listens
- The executable name of the gateway that the Oracle Net Listener starts in response to incoming connection requests

Syntax of listener.ora File Entries

The Oracle database communicates with the gateway using Oracle Net and any supported protocol adapters. The following syntax of the address on which the Oracle Net Listener listens using the TCP/IP protocol adapter::

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

Where:

Variable	Description
host_name	is the name of the machine on which the gateway is installed.
port_number	specifies the port number used by the Oracle Net Listener. If you have other listeners running on the same machine, then the value of *port_number* must be different from the other listeners' port numbers.

To direct the Oracle Net Listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file.

Note: You must use the same SID value in the `listener.ora` file and the `tnsnames.ora` file which will be configured in the next step.

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

Where:

Variable	Description
gateway_sid	specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.
oracle_home_directory	specifies the Oracle home directory where the gateway resides.
`hsodbc`	specifies the executable name of the Oracle Database Gateway for ODBC.

If you already have an existing Oracle Net Listener, then add 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=hsodbc)
   )
)
```

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

Stop and Start the Oracle Net Listener for the Gateway

You must stop and restart the Oracle Net Listener to initiate the new settings, as follows:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. If the service is already running, click **Stop** to stop it.
4. Click **Start** to start or restart the service.

Configure the Oracle Database for Gateway Access

Before you use the gateway to access an ODBC data source you must configure the Oracle database to enable communication with the gateway over Oracle Net.

To configure the Oracle database you must add connect descriptors to the `tnsnames.ora` file. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database is installed. You cannot use the Oracle Net Assistant or the Oracle Net 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.

Configuring tnsnames.ora

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is a syntax of the Oracle Net entry using the TCP/IP protocol:

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

Where:

Table 11–1 Gateway Parameters for tnsnames.ora File

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as `hsodbc`. 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`	is the TCP protocol used for TCP/IP connections.
host_name	specifies the machine where the gateway is running.
port_number	matches the port number used by the Oracle Net Listener that is listening for the gateway. The Oracle Net Listener's port number can be found in the `listener.ora` file used by the Oracle Net Listener. See "Syntax of listener.ora File Entries" on page 11-3.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the Oracle Net Listener that is listening for the gateway. See "Configure Oracle Net Listener for the Gateway" on page 11-3 for more information.
`(HS=OK)`	specifies that this connect descriptor connects to a non-Oracle system.

Create Database Links

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

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 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 ODBC data source.

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.

To access the ODBC data source, you must create a database link. A public database link is the most common of database links.

```
SQL> CREATE PUBLIC DATABASE LINK dblink CONNECT TO
  2  "user" IDENTIFIED BY "password" USING 'tns_name_entry';
```

Where:

Variable	Description
dblink	is the complete database link name.
tns_name_entry	specifies the Oracle Net connect descriptor specified in the `tnsnames.ora` file that identifies the gateway

After the database link is created you can verify the connection to the ODBC data source, as follows:

```
SQL> SELECT * FROM DUAL@dblink;
```

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

Encrypt Gateway Initialization Parameter Values

The gateway uses user IDs and passwords to access the information in the remote database. Some user IDs 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 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.

Configure the Gateway to Access Multiple ODBC Data Sources

The tasks for configuring the gateway to access multiple ODBC data sources are similar to the tasks for configuring the gateway for a single data source. The configuration example assumes the following:

- The gateway is installed and configured with the SID of `hsodbc`.
- The gateway is configured to access one ODBC data source named `dsn1`.
- Two ODBC data sources named dsn2 and dsn3 where dsn2 and dsn3 are the names of the system DSN defined in the Windows ODBC Data Source Administrator, are being added.

Multiple ODBC Data Sources Example: Configuring the Gateway

Choose One System ID for Each ODBC Data Source

A separate instance of the gateway is needed for each ODBC data source. Each instance needs its own gateway System ID (SID). For this example, the gateway SIDs are chosen for the instances that access the ODBC data source:

- `hsodbc2` for the gateway accessing data source `dsn2`.
- `hsodbc3` for the gateway accessing data source `dsn3`.

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*`\hs\admin\inithsodbc.ora`, twice, naming one with the gateway SID for `dsn2` and the other with the gateway SID for `dsn3`:

```
> cd ORACLE_HOME\hs\admin
> copy inithsodbc.ora inithsodbc2.ora
> copy inithsodbc.ora inithsodbc3.ora
```

Change the value of the `HS_FDS_CONNECT_INFO` parameter in the new files as follows:

For `inithsodbc2.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=dsn2
```

For `inithsodbc3.ora`, enter the following:

```
HS_FDS_CONNECT_INFO=dsn3
```

Note: If you have multiple gateway SIDs for the same ODBC data source 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 ODBC Data Sources Example: Configuring Oracle Net Listener

Add Entries to listener.ora

Add two new entries to the Oracle Net 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=hsodbc)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=hsodbc)
   )
   (SID_DESC=
      (SID_NAME=hsodbc2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=hsodbc)
   )
   (SID_DESC=
      (SID_NAME=hsodbc3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=hsodbc)
   )
)
```

where, `oracle_home_directory` is the directory where the gateway resides.

Multiple ODBC Data Sources Example: Stopping and Starting the Oracle Net Listener

Perform the following steps:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

Multiple ODBC Data Sources Example: Configuring Oracle Database for Gateway Access

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

```
old_dsn_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=hsodbc))
                (HS=OK))
new_dsn2_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=hsodbc2))
                 (HS=OK))
new_dsn3_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
```

```
(HOST=host_name))
(CONNECT_DATA=
    (SID=hsodbc3))
(HS=OK))
```

The value for `PORT` is the TCP/IP port number of the Oracle Net Listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the Oracle Net 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 Oracle Net Listener.

Multiple ODBC Data Sources Example: Accessing ODBC Data

Enter the following to create a database link for the `hsodbc2` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK ODBC2 CONNECT TO
  2  "user2" IDENTIFIED BY "password2" USING 'new_dsn2_using';
```

Enter the following to create a database link for the `hsodbc3` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK ODBC3 CONNECT TO
  2  "user3" IDENTIFIED BY "password3" USING 'new_dsn3_using';
```

After the database links are created, you can verify the connection to the new ODBC data sources, as in the following:

```
SQL> SELECT * FROM ALL_USERS@ODBC2;

SQL> SELECT * FROM ALL_USERS@ODBC3;
```

Part VII

Installing and Configuring Oracle Database Gateway for DRDA

Part VII, "Installing and Configuring Oracle Database Gateway for DRDA" describes how to install and configure of Oracle Database Gateway for DRDA.

It contains the following chapters:

- Chapter 12, "Installing Oracle Database Gateway for DRDA"
- Chapter 13, "Configuring the DRDA Server"
- Chapter 14, "Configuring Oracle Database Gateway for DRDA"
- Chapter 15, "Security Considerations"
- Chapter 16, "Migration From Previous Releases"

12

Installing Oracle Database Gateway for DRDA

This chapter guides you through the installation procedure of Oracle Database Gateway for DRDA.

To install the gateway, follow these steps:

1. Ensure that the system meets all of the hardware and software requirements specified in "System Requirements for Oracle Database Gateway for DRDA" on page 12-1.
2. Log on to your host computer as a member of the Administrators group.
3. Run the Oracle Universal Installer.

   See "Step Through the Oracle Universal Installer" section on page 12-3 for more information about running the Oracle Universal Installer.

   Oracle Universal Installer is a menu-driven utility that guides you through the installation of the gateway by prompting you with action items. The action items and the sequence in which they appear depend on your platform.

   See Table 12–2 for a description of the installation procedure of Oracle Database Gateway for DRDA.

System Requirements for Oracle Database Gateway for DRDA

This section provides information about the hardware and software requirements for the gateway. It contains the following sections:

Hardware Requirements

Table 12–1 lists the minimum hardware requirements for Oracle Database Gateway for DRDA.

Table 12–1 Hardware Requirements for Oracle Database Gateway for DRDA

Requirement	Value
Disk space for Oracle home	800 MB
Disk space for system drive (*SYSTEM_DRIVE*:\)	260 MB

Table 12–1 (Cont.) Hardware Requirements for Oracle Database Gateway for DRDA

Requirement	Value
Temporary disk space	125 MB
Physical Memory	64 MB of real memory is the recommended minimum for running one instance of the gateway. Running additional instances of the Oracle Database Gateway for DRDA might require additional real memory or increased swap space to achieve reasonable performance. The total real memory requirement for each concurrent use of the gateway depends on the following factors: ■ Number of concurrent TCP/IP connections opened by each user ■ Number of data items being transferred between the gateway and the remote transaction program ■ Additional factors such as configured network buffer size
Processor	This gateway requires a host with an Intel or 100% compatible PC with a Pentium-based processor that can run the required version of Microsoft Windows.

Checking the Hardware Requirements

To ensure that the system meets the minimum requirements, follow these steps:

1. Determine the physical RAM size. For a computer using Windows 2000, for example, open **System** in the control panel and select the **General** tab. If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.
2. Determine the size of the configured swap space (also known as paging file size). For a computer using Windows 2000, for example, open **System** in the control panel, select the **Advanced** tab, and click **Performance Options**.

   If necessary, then see your operating system documentation for information about how to configure additional swap space.
3. Determine the amount of free disk space on the system. For a computer using Windows 2000, for example, open **My Computer**, right-click the drive where the Oracle software is to be installed, and select **Properties**.
4. Determine the amount of disk space available in the `temp` directory. This is equivalent to the total amount of free disk space, minus what will be needed for the Oracle software to be installed.

   If there is less than 125 MB of disk space available in the `temp` directory, then first delete all unnecessary files. If the temp disk space is still less than 125 MB, then set the `TEMP` or `TMP` environment variable to point to a different hard drive. For a computer using Windows 2000, for example, open the **System** control panel, select the **Advanced** tab, and click **Environment Variables**.

Software Requirements

Oracle Database Gateway for DRDA is supported on the following Microsoft Windows operating systems:

- Microsoft Windows 2000 Server (with Service Pack 2 or later)
- Microsoft Windows 2000 Professional (with Service Pack 2 or later)
- Microsoft Windows 2003 Server

- Microsoft Windows XP Professional

Certified Configurations

Oracle continually updates supported gateway configurations. For the latest supported configuration information, visit the OTN Web site:

http://otn.oracle.com/products/gateways/pdf/mainframe_certification.pdf

Step Through the Oracle Universal Installer

Table 12–2 describes the installation procedure for Oracle Database Gateway for DRDA.

Table 12–2 The Oracle Universal Installer: Steps for Installing the Gateway

Screen	Response
Oracle Universal Installer: Welcome	Click **Next**.
Oracle Universal Installer: Specify Home Details	Specify a name for the installation in the Name field. You can also choose not to edit the default setting of the Name field of the Specify Home Details screen. The Path field in the Specify Home Details screen is where you specify the destination for your installation. You need not edit the path specification in the Path field. The default setting for this field points to *ORACLE_HOME*. After you set the fields in the Specify Home Details screen as necessary, click **Next** to continue. After loading the necessary information from the installation, the Oracle Universal Installer displays the Available Products screen.
Oracle Universal Installer: Available Product Components	**a.** Select **Oracle Database Gateway for DRDA 11.1.0.5.0**. **b.** Click **Next**.
Oracle Universal Installer: Summary	The Installation Summary screen enables you to review a tree list of options and components for this installation. Click **Install** to start installation.
Oracle Net Configuration Assistant: Welcome	Click **Cancel**.
Oracle Net Configuration Assistant:	Click **Yes**.
Oracle Universal Installer: Configuration Tools	Click **Exit**.
Exit	The final screen of the Oracle Universal Installer is the End of Installation screen. Click **Exit** to exit the installer.

The gateway is now installed.

When the Oracle Universal Installer confirms that the installation is complete, verify that the installation procedure was successful. To do this, read the contents of the installation log file, which is located in the `C:\Program Files\Oracle\Inventory\logs` directory.

The default file name is `InstallActions`*`YYYY-MM-DD_HH-mm-SS-AM/PM`*`.log`, where:

YYYY is year
MM is month
DD is day
HH is hour

mm is minute
SS is seconds
AM/PM is daytime or evening

Each of these variables in the log file name represents the date and time the product was installed.

13

Configuring the DRDA Server

This chapter describes tasks you must perform to configure the DRDA server. Each supported operating system is addressed separately. Experience with the given operating system and database is required.

The steps for configuring your remote DRDA server apply to the following DRDA servers:

- DB2/OS390
- DB2/400
- DB2/UDB

Configuring a DRDA database to enable access by the gateway requires actions on the DRDA database and on certain components of the host operating system. Although no Oracle software is installed on the host system, access to, and some knowledge of, the host system and DRDA database are required during the configuration. Refer to the vendor documentation for complete information about your host system and DRDA database.

This chapter contains the following sections:

Configuring the DRDA Server for DB2/OS390

Perform the following tasks to configure the DRDA server with DB2 on an OS390 system:

1. Define the user ID that owns the package

   During gateway configuration, you will need to run the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password that are used when the procedure is run (either implied as the current Oracle user or explicitly defined in the `CREATE DATABASE LINK` command) must have proper authority on the DRDA Server to create the package. This same user ID should be used to create and to own the ORACLE2PC (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have one or more of the following privileges on the DRDA Server:

   - Package privileges of `BIND`, `COPY`, and `EXECUTE`, for example:

     ```
     GRANT BIND    ON PACKAGE drda1.* TO userid
     ```

```
GRANT COPY    ON PACKAGE drda1.* TO userid
GRANT EXECUTE ON PACKAGE drda1.* TO PUBLIC
```

- Collection privilege of `CREATE IN`, for example:

```
GRANT CREATE IN ON PACKAGE drda1 TO USER userid
```

- System privileges of `BINDADD` and `BINDAGENT`, for example:

```
GRANT BINDADD   TO USER userid
GRANT BINDAGENT TO USER userid
```

- Database privilege of `CREATETAB`, for example:

```
GRANT CREATETAB ON DATABASE database TO USER userid
```

Choose a user ID now that will own the package and the `ORACLE2PC` table. Ensure that this user ID is defined to both DB2 and OS/390 (MVS).

2. Define the recovery user ID

   During gateway configuration, the recovery user ID and password are specified in the gateway initialization file using the `DRDA_RECOVERY_USERID` and `DRDA_RECOVERY_PASSWORD` parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password that are defined in these parameters. This user ID must have execute privileges on the package and must be defined to the DRDA database. If the user ID is not specified in `DRDA_RECOVER_USERID`, then the gateway attempts to connect to a user ID of `ORARECOV` when a distributed transaction is in doubt.

   Determine the user ID and password that you will use for recovery.

3. Determine DRDA location name for DB2 instance

   The DRDA location name is required as a gateway parameter. To determine the location name, run the SQL query from a DB2 SPUFI session:

```
SELECT CURRENT SERVER FROM any_table
```

   where *any_table* is a valid table with one or more rows.

   If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact the system administrator to arrange to set a location name for the instance.

4. Configure DB2 Distributed Data Facility for Gateway

   DB2 DDF is the component of DB2 that manages all distributed database operations, both DRDA and non-DRDA.

   If your site uses DB2 distributed operations, then DDF is probably operational on the DB2 instance that you plan to access through the gateway. If DDF is not operational, then you must configure it and start it as described in the appropriate DB2 documentation.

   Even if DDF is operational on the DB2 instance, it might be necessary to make changes to the DDF Communication Database (CDB) tables to specify the authorization conduct of DRDA sessions from the gateway. This can be done by properly authorized users with a utility such as the DB2 SPUFI utility. If you make changes to CDB tables, then you must stop and restart DDF for the changes to take effect. Refer to Chapter 15, "Security Considerations", for additional CDB tables and security information.

Configuring the DRDA Server for DB2/400

Experience with DB2/400 and AS/400 is required to perform the following steps:

1. Define the user ID that owns the package

   During gateway configuration, you will need to run the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password that are used when the procedure is run (either implied as the current Oracle user or explicitly defined in the `CREATE DATABASE LINK` command) must have proper authority on the DRDA Server to create the package. This same user ID should be used to create and to own the `ORACLE2PC` (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have the following privileges on the DRDA Server:

   - Use authority on the `CRTSQLPKG` command
   - Change authority on the library in which the package will be created

   Choose a user ID now that will own the package and the `ORACLE2PC` table. Ensure that this user ID is defined to DB2/400 and AS/400.

2. Define the recovery user ID

   During gateway configuration, the recovery user ID and password are specified in the gateway initialization file using the `DRDA_RECOVERY_USERID` and `DRDA_RECOVERY_PASSWORD` parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password that are defined in these parameters. This user ID must have execute privileges on the package and must be defined to the DRDA database. If the user ID is not specified in `DRDA_RECOVER_USERID`, then the gateway attempts to connect to a user ID of `ORARECOV` when a distributed transaction is in doubt.

   Determine the user ID and password that you will use for recovery.

3. Determine DRDA location name for DB2/400 instance

   The DRDA location name is required as a gateway parameter. To determine the location name, run the following SQL query from a STRSQL session. If SQL is unavailable on the system, then use the AS/400 command `DSPRDBDIRE` to identify your "LOCAL" DRDA Server.

   ```
   SELECT CURRENT SERVER FROM any_table
   ```

   where *any_table* is a valid table with one or more rows.

   If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact the system administrator to arrange to set a location name for the instance.

Configuring the DRDA Server for DB2/UDB (Universal Database)

Experience with DB2/UDB, configuring the communication subsystem of DB2/UDB, and the host System Administration tools is required to perform the following steps.

1. Define the user ID that owns the package

   During gateway configuration, you will need to run the Bind Package Stored Procedure to bind the gateway package on the DRDA Server. To properly bind the package, the user ID and password that are used when the procedure is run (either implied as the current Oracle user or explicitly defined in the `CREATE DATABASE LINK` command) must have proper authority on the DRDA Server to

create the package. This same user ID should be used to create and to own the ORACLE2PC (two-phase commit) table. The user ID that is used to bind or rebind the DRDA package must have one or more of the following privileges on the DRDA Server:

- Package privileges of BIND and EXECUTE, for example:

```
GRANT BIND    ON PACKAGE drda1.g2drsql TO USER userid
GRANT EXECUTE ON PACKAGE drda1.g2drsql TO PUBLIC
```

- Schema privileges of CREATEIN, for example:

```
GRANT CREATEIN ON SCHEMA otgdb2 TO USER userid
GRANT CREATEIN ON SCHEMA drda1 TO USER userid
```

- Database authorities of CONNECT, BINDADD, and CREATETAB, for example:

```
GRANT CONNECT   ON DATABASE TO USER userid
GRANT BINDADD   ON DATABASE TO USER userid
GRANT CREATETAB ON DATABASE TO USER userid
```

Now choose a user ID that will own the package and ORACLE2PC table. Ensure that this user ID is defined to both the DB2 instance ID and the operating system.

2. Define the recovery user ID

   During gateway configuration, the recovery user ID and password are specified in the gateway initialization file using the DRDA_RECOVERY_USERID and DRDA_RECOVERY_PASSWORD parameters. If a distributed transaction fails, then the recovery process connects to the remote database using the user ID and password that are defined in these parameters. This user ID must have execute privileges on the package and must be defined to the DRDA database. If the user ID is not specified in DRDA_RECOVER_USERID, then the gateway attempts to connect to a user ID of ORARECOV when a distributed transaction is in doubt.

   Determine the user ID and password that you will use for recovery.

3. Determine DRDA location name for DB2/UDB instance

   The DRDA location name is required as a gateway parameter. To determine the location name, run the SQL query from a DB2 CLI session:

```
SELECT CURRENT SERVER FROM any_table
```

   where *any_table* is a valid table with one or more rows.

   If the value returned by this query is blank or null, then the DRDA location name has not been established. Contact your system administrator to arrange to set a location name for the instance.

14

Configuring Oracle Database Gateway for DRDA

After installing the gateway, perform the following tasks to configure Oracle Database Gateway for DRDA:

1. Configure the Gateway Initialization Parameter File
2. Configure Oracle Net for the Gateway
3. Configure Two-Phase Commit
4. Bind the DRDA Gateway Package
5. Create Tables and Views for Data Dictionary Support
6. Grant Authority to the DRDA Package
7. Configure the Oracle Database for Gateway Access
8. Create Database Links
9. Configure the Gateway to Access Multiple DRDA Databases

SQL scripts are provided to perform steps such as creating the `ORACLE2PC` table, removing obsolete tables and views, using previous releases, and creating tables and views to provide data dictionary support.

These scripts must be run on the DRDA Server platform using a database native tool (such as SPUFI on DB2/OS390), because no tool is provided with the gateway to run these scripts. Note that when running these scripts, the user ID used must be suitably authorized.

Choose the appropriate subdirectory for your DRDA server platform from the following list:

- for DB2/OS390: choose `tg4drda\install\db2`
- for DB2/400: choose `tg4drda\install\as400`
- for DB2/UDB: choose `tg4drda\install\db2udb`

Configure the Gateway Initialization Parameter File

Perform the following tasks to configure the gateway initialization parameter file

1. Choose a System Identifier for the Gateway
2. Customize the Initialization Parameter File

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 DRDA database you are accessing. However, if you want to access two DRDA databases, you need two gateway SIDs, one for each instance of the gateway. If you have one DRDA 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 DRDA database. The SID is used as part of the file name for the initialization parameter file.

Customize the Initialization Parameter File

The data in this chapter describes the configuration process for the gateway. You should notice that most, if not all, gateway parameters have been moved into the init*sid*.ora initialization file. To configure the host for the Oracle Database Gateway for IBM DRDA, you tailor the parameter files for your installation.

Note: In previous versions of the gateway, the initialization parameters were stored in the files named init*sid*.ora and init*sid*.gtwboot in the gateway instance directories. With Release 10.1.0.2.0 of the gateway, most parameters that were in 'init*sid*.gtwboot' have been moved to init*sid*.ora. The syntax of the init*sid*.ora has been simplified. Refer to Appendix C, "Initialization Parameters" for details.

When migrating from previous releases of TG4DRDA, be aware of these differences.

---

1. Copy the gateway initialization

   Sample gateway initialization files (init*sid*.ora) are shipped on the distribution media. These files are in the *ORACLE_HOME*\tg4drda\admin directory:

   - initDB2.ora, for DB2/OS390 remote servers
   - initAS400.ora, for DB2/400 remote servers
   - initDB2UDB.ora, for DB2/UDB remote servers

   Copy one of these sample files into the same directory, renaming it with the name of your gateway SID. For example, if you chose your SID to be DRD1, and if your remote server is DB2, then copy the initDB2.ora file and rename it initDRD1.ora.

2. Determine settings for gateway initialization parameters

   Your Configuration Worksheet in Appendix D, "Configuration Worksheet for DRDA" should be complete. If not, review the incomplete entries and refer to the sections listed for more information. You need this information to tailor the gateway initialization file, init*sid*.ora.

   Refer to Appendix C, "Initialization Parameters" for information on the DRDA-specific init*sid*.ora parameters.

   Following are the parameters:

   - Required Parameters

When you edit your `initsid.ora` file, you must change the values of all the parameters listed in the Configuration Worksheet in Appendix D, "Configuration Worksheet for DRDA", using the values in the right-hand column of the worksheet.

You will also need to set certain Globalization Support gateway parameters. For more information on setting these parameters, refer to Appendix E, "Globalization Support for DRDA".

- Optional Parameters

  Several DRDA-specific parameters are not required, but you might want to change them. Unless otherwise indicated, these parameters are described in Appendix C, "Initialization Parameters".

Table 14–1 Optional DRDA-Specific Parameters

DRDA parameters	Description
DRDA_DISABLE_CALL	Used to disable stored procedure support for DRDA Servers on which the gateway does not support stored procedures
DRDA_ISOLATION_LEVEL	Defines the package Isolation Level
HS_DB_NAME	Specifies the database SID name and must be set to the gateway SID
DRDA_OPTIMIZE_QUERY	Used for data query optimization
DRDA_PACKAGE_COLLID	Defines the package collection ID
DRDA_PACKAGE_NAME	Defines the name of the package
DRDA_PACKAGE_OWNER	Defines the owner of the package. By default, the owner is the user ID that is used when you run the g4drutl bind utility. This parameter is not valid for SQL/DS.
DRDA_PACKAGE_SECTIONS	Defines the maximum number of concurrent OPEN cursors at the remote server
HS_DB_DOMAIN	Specifies the gateway database domain

The values that are set in your init*sid*.ora file should work for most installations. Edit the values if changes are needed. For information on Globalization Support-related init*sid*.ora parameters, refer to "Globalization Support Parameters in the Gateway Initialization File" on page E-4.

3. Tailor the `initsid.ora` File

   After you have copied the sample initialization file, you will need to tailor it to your installation. While many parameters can be left to their defaults, some parameters must be changed for correct operation of the gateway. Give attention to the following DRDA and HS parameters. Also, give attention to the security aspects of the initialization file. Chapter 15, "Security Considerations", contains details on encryption of passwords that would otherwise be embedded in the initialization file. See Appendix D, "Configuration Worksheet for DRDA" for a description of each parameter:

   - DRDA_CONNECT_PARM
   - DRDA_PACKAGE_COLLID
   - DRDA_PACKAGE_NAME
   - DRDA_PACKAGE_OWNER

- DRDA_REMOTE_DB_NAME
- FDS_CLASS
- HS_DB_NAME
- HS_DB_DOMAIN

Configure Oracle Net for the Gateway

The gateway requires Oracle Net to communicate with the Oracle database. After configuring the gateway, perform the following tasks to configure Oracle Net to work with the gateway:

1. Configure Oracle Net Listener for the Gateway
2. Stop and Start the Oracle Net Listener for the Gateway

Configure Oracle Net Listener for the Gateway

The Oracle Net Listener listens for incoming requests from the Oracle database. For the Oracle Net Listener to listen for the gateway, information about the gateway must be added to the Oracle Net Listener configuration file, `listener.ora`. This file by default is located in *ORACLE_HOME*`\network\admin`, where *ORACLE_HOME* is the directory under which the gateway is installed.

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

- A list of Oracle Net addresses on which the Oracle Net Listener listens
- The executable name of the gateway that the Oracle Net Listener starts in response to incoming connection requests

Syntax of listener.ora File Entries

The Oracle database communicates with the gateway using Oracle Net and any supported protocol adapters. The following syntax of the address on which the Oracle Net Listener listens using the TCP/IP protocol adapter:

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

Where:

Variable	Description
host_name	is the name of the machine on which the gateway is installed.
port_number	specifies the port number used by the Oracle Net Listener. If you have other listeners running on the same machine, then the value of *port_number* must be different from the other listeners' port numbers.

To direct the Oracle Net Listener to start the gateway in response to incoming connection requests, add an entry to the `listener.ora` file.

Note: You must use the same SID value in the `listener.ora` file and as the `tnsnames.ora` file which will be configured in the next step.

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
         (SID_NAME=gateway_sid)
         (ORACLE_HOME=oracle_home_directory)
         (PROGRAM=g4drsrv)
      )
   )
```

Where:

Variable	Description
gateway_sid	specifies the SID of the gateway and matches the gateway SID specified in the connect descriptor entry in the `tnsnames.ora` file.
oracle_home_directory	specifies the Oracle home directory where the gateway resides.
`g4drsrv`	specifies executable name of the Oracle Database Gateway for DRDA.

If you already have an existing Oracle Net Listener, then add 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=g4drsrv)
   )
)
```

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

Stop and Start the Oracle Net Listener for the Gateway

You must stop and restart the Oracle Net Listener to initiate the new settings, as follows:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. If the service is already running, click **Stop** to stop it.
4. Click **Start** to start or restart the service.

Configure Two-Phase Commit

Depending on the server you are connecting to, run the appropriate script from the following:

- If connecting to DB2/UDB, the perform the following steps to create the ORACLE2PC table:

  1. Log in to the system where DB2/UDB is running.

     Check that you have the ability to address the DB2/UDB instance where the ORACLE2PC table will reside.

  2. Copy the following files.

     Copy from the *ORACLE_HOME*\tg4drda\install\db2udb directory:

     - o2pc.sql (SQL script for creating the table)
     - o2pcg.sql (SQL script for granting package access to PUBLIC)

  3. Connect to the database.

     Use the user ID that you will use for binding the package:

     ```
     $ db2 'CONNECT TO database USER userid USING password'
     ```

     Note: The user ID must have CONNECT, CREATETAB, and BINDADD authority to be able to connect to the database, to create the table, and to create the package.

     For more information, refer to "Configuring the DRDA Server for DB2/UDB (Universal Database)" on page 13-3.

  4. Create the ORACLE2PC table:

     ```
     $ db2 -tf o2pc.sql
     ```

  5. Commit the transaction:

     ```
     $ db2 'COMMIT'
     ```

  6. Optionally, verify that the table was created under the correct user ID:

     ```
     $ db2 'LIST TABLES FOR USER'
     $ db2 'COMMIT'
     ```

  7. Disconnect from the session:

     ```
     $ db2 'DISCONNECT CURRENT'
     ```

- If connecting to DB2/400, then run the following SQL script to create your ORACLE2PC table:

  ORACLE_HOME\tg4drda\install\as400\o2pc.sql

- If connecting to DB2/OS390, then run the following SQL script to create your ORACLE2PC table:

  ORACLE_HOME\tg4drda\install\db2\o2pc.sql

Check preceding codes. Codes adapted from UNIX guide.

Bind the DRDA Gateway Package

The product requires a package to be bound on the DRDA server. The gateway has an internal, stored procedure that must be used to create this package. The internal stored procedure is called from an Oracle database. Before this package can be bound on the DRDA server, the gateway initialization file must be correctly configured. Refer to Appendix D, "Configuration Worksheet for DRDA" for more information.

DRDA Gateway Package Binding Considerations

Before binding the DRDA gateway package, perform the following steps:

1. Check the user has the appropriate privileges

   The DRDA package must be bound with the internal Stored Procedure `GTW$_BIND_PKG`. The user ID used to bind the DRDA package must have the appropriate privileges on the remote database, as described in Chapter 13, "Configuring the DRDA Server".

2. Check all DRDA parameter settings

   Check all DRDA parameter settings to be sure that they are set correctly before you start the bind. For example, the default for `DRDA_DISABLE_CALL` only works if your DRDA database supports stored procedures. If not, then you must change the setting. Also, the value for `DRDA_PACKAGE_NAME` must be unique if you have any older versions of the gateway installed. New packages replace any old packages with the same name, causing versions of the gateway that use the old package to fail. Refer to Appendix C, "Initialization Parameters" for information on the parameters and their settings.

DRDA Gateway Package Binding Steps

1. Log on to an Oracle database

   Use SQL*Plus:

   ```
   > sqlplus system/manager
   ```

2. Create a database link

   Create a database link with a user ID and with a password that has proper authority on the DRDA Server to create packages.

   ```
   SQL> CREATE PUBLIC DATABASE LINK dblink
   2 CONNECT TO "userid" IDENTIFIED BY "password"
   3 USING 'tns_name_entry'
   ```

   Note: The user ID that is creating the public database link must have the `CREATE PUBLIC DATABASE LINK` privilege.

3. Run the stored procedure `GTW$_BIND_PKG`

   ```
   SQL> exec GTW$_BIND_PKG@dblink;
   SQL> COMMIT;
   ```

   This creates and commits the package. If any errors are reported, then correct the gateway initialization file parameters as needed.

Create Tables and Views for Data Dictionary Support

Perform the following steps to create the data dictionary tables and views:

1. Upgrading from a previous gateway release

   If you are upgrading from a previous version of the gateway then run the `dropold.sql` script to drop the old data dictionary definitions.

2. Creating the data dictionary tables

   Run the `g4ddtab.sql` script to create the data dictionary tables.

3. Creating the data dictionary views

   Run the following `g4ddvwXX.sql` scripts to create the data dictionary views:

   For DB2/OS390 V7 (RACF security): `g4ddvwr7.sql`

   For DB2/OS390 V7 (DB2 security): `g4ddvws7.sql`

   For DB2/OS390 V8 (RACF security): `g4ddvwr8.sql`

   For DB2/OS390 V8 (DB2 security): `g4ddvws8.sql`

   For DB2/400 V5.1: `g4ddvw51.sql`

   For DB2/400 V5.2: `g4ddvw52.sql`

   For DB2/UDB V7: `g4ddvwu7.sql`

   For DB2/UDB V8: `g4ddvwu8.sql`

Grant Authority to the DRDA Package

To grant all users access to use the `ORACLE2PC` table, run the appropriate script from the following:

- If server is DB2/UDB, then run

  ORACLE_HOME`\tg4drda\install\db2udb\o2pcg.sql`

- If server is DB2/OS390, then run

  ORACLE_HOME`\tg4drda\install\db2\o2pcg.sql`

- If server is DB2/400, then run

  ORACLE_HOME`\tg4drda\install\as400\o2pcg.sql`

Preceding codes need to be verified.

Configure the Oracle Database for Gateway Access

Before you use the gateway to access DRDA data you must configure the Oracle database to enable communication with the gateway over Oracle Net.

To configure the Oracle database you must add connect descriptors to the `tnsnames.ora` file. By default, this file is in `$ORACLE_HOME/network/admin`, where `$ORACLE_HOME` is the directory in which the Oracle database is installed. You cannot use the Oracle Net Assistant or the Oracle Net 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.

Configuring tnsnames.ora

Edit the `tnsnames.ora` file to add a connect descriptor for the gateway. The following is a syntax of the Oracle Net entry using the TCP/IP protocol.

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

Where:

Table 14–2 Gateway Parameters for tnsnames.ora File

Variable	Description
connect_descriptor	is the description of the object to connect to as specified when creating the database link, such as `tg4drda`. 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	is the TCP protocol used for TCP/IP connections.
host_name	specifies the machine where the gateway is running.
port_number	matches the port number used by the Oracle Net Listener that is listening for the gateway. The Oracle Net Listener's port number can be found in the `listener.ora` file used by the Oracle Net Listener. See "Syntax of listener.ora File Entries" on page 14-4.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the Oracle Net Listener that is listening for the gateway. See "Configure Oracle Net Listener for the Gateway" on page 14-4 for more information.
`(HS=OK)`	specifies that this connect descriptor connects to a non-Oracle system.

Create Database Links

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

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 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 DRDA 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.

To access the DRDA server, you must create a database link. A public database link is the most common of database links.

```
SQL> CREATE PUBLIC DATABASE LINK dblink CONNECT TO
  2  "user" IDENTIFIED BY "password" USING 'tns_name_entry';
```

Where:

Variable	Description
dblink	is the complete database link name.
tns_name_entry	specifies the Oracle Net connect descriptor specified in the `tnsnames.ora` file that identifies the gateway

After the database link is created you can verify the connection to the DRDA database, as follows:

```
SQL> SELECT * FROM DUAL@dblink;
```

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

Configure the Gateway to Access Multiple DRDA Databases

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

- The gateway is installed.
- The gateway is configured for one DRDA database named `db1`.
- Two DRDA databases named `db2` and `db3` on a host with IP Address 204.179.79.15 are being added.

Multiple DRDA Databases Example: Configuring the Gateway

Choose One System ID for Each DRDA Database

A separate instance of the gateway is needed for each DRDA database. Each instance needs its own gateway System ID (SID). For this example, the gateway SIDs are chosen for the instances that access the DRDA databases:

- `tg4drda2` for the gateway accessing database `db2`.
- `tg4drda3` 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*`\tg4drda\admin\inittg4drda.ora`, twice, naming one with the gateway SID for `db2` and the other with the gateway SID for `db3`:

```
> cd ORACLE_HOME\tg4drda\admin
> copy inittg4drda.ora inittg4drda2.ora
> copy inittg4drda.ora inittg4drda3.ora
```

Note: If you have multiple gateway SIDs for the same DRDA 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 DRDA Databases Example: Configuring Oracle Net Listener

Add Entries to listener.ora

Add two new entries to the Oracle Net 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=tg4drda)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=g4drsrv)
   )
   (SID_DESC=
      (SID_NAME=tg4drda2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=g4drsrv)
   )
   (SID_DESC=
      (SID_NAME=tg4drda3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=g4drsrv)
   )
)
```

where, `oracle_home_directory` is the directory where the gateway resides.

Multiple DRDA Databases Example: Stopping and Starting the Oracle Net Listener

Perform the following steps:

1. From the **Start** menu, select **Settings**, **Control Panel** and then select **Services**.
2. Select the Oracle Net Listener service for the gateway.
3. Click **Stop**.
4. Click **Start**.

Multiple DRDA Databases Example: Configuring Oracle Database for Gateway Access

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.

This example describes how to configure Oracle Net on the Oracle database for multiple gateway instances. It 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=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4drda))
                (HS=OK))
new_db2_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4drda2))
                 (HS=OK))
new_db3_using=(DESCRIPTION=
               (ADDRESS=
                 (PROTOCOL=TCP)
                 (PORT=port_number)
                 (HOST=host_name))
                 (CONNECT_DATA=
                     (SID=tg4drda3))
                 (HS=OK))
```

The value for `PORT` is the TCP/IP port number of the Oracle Net Listener that is listening for the gateway. The number can be found in the `listener.ora` file used by the Oracle Net 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 Oracle Net Listener.

Multiple DRDA Databases Example: Accessing DRDA Data

Enter the following to create a database link for the `tg4drda2` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK DRDA2 CONNECT TO
  2  "user2" IDENTIFIED BY "password2" USING 'new_db2_using';
```

Enter the following to create a database link for the `tg4drda3` gateway:

```
SQL> CREATE PUBLIC DATABASE LINK DRDA3 CONNECT TO
  2  "user3" IDENTIFIED BY "password3" USING 'new_db3_using';
```

After the database links are created, you can verify the connection to the new DRDA databases, as in the following:

```
SQL> SELECT * FROM ALL_USERS@DRDA2;

SQL> SELECT * FROM ALL_USERS@DRDA3;
```

15

Security Considerations

The gateway architecture involves multiple computer setups that have distinct security capabilities and limitations. This chapter provides information for planning and implementing your security system.

It contains the following sections:

Security Overview

When you connect several different systems, generally the system with the strictest security requirements dictates and rules the system.

Gateway security involves two groups:

- Users and applications that are permitted access to a given gateway instance and DRDA database server
- Server database objects that users and applications are able to query and update

You can control access in the gateway architecture at several points. Control over database object access is provided by each DRDA database server with `GRANT`s and related native authorization mechanisms based on user ID.

When the gateway is involved in a SQL request, security mechanisms are in effect for each DRDA system component encountered by the gateway. The first system component encountered is the application tool or 3GL program. The last system component encountered is the DRDA database.

Authenticating Application Logons

An application must connect to an Oracle database before using the gateway. The type of logon authentication that you use determines the resulting Oracle user ID and can affect gateway operation. There are two basic types of authentication:

- Oracle authentication: With Oracle authentication, each Oracle user ID has a password known to Oracle database. When an application connects to the server, it supplies a user ID and password. Oracle database confirms that the user ID exists and that the password matches the one kept in the database.

- Operating system authentication: With operating system authentication, the servers underlying operating system is responsible for authentication. An Oracle user ID that is created with the `IDENTIFIED EXTERNALLY` attribute, instead of a password, is accessed with operating system authentication. To log into such a user ID, the application supplies a forward slash (`/`) for a user ID and does not supply a password.

  To perform operating system authentication, the server determines the requester's operating system user ID, optionally adds a fixed prefix to it, and uses the result as the Oracle user ID. The server confirms that the user ID exists and is `IDENTIFIED EXTERNALLY`, but no password checking is done. The underlying assumption is that users were authenticated when they logged into the operating system.

  Operating system authentication is not available on all platforms and is not available in some `Oracle Net` (client-server) and multi-threaded server configurations. Refer to the *Oracle Database Installation Guide 11g for UNIX Systems* and Oracle Net documentation to determine the availability of this feature.

For more information about authenticating application logons, refer to the *Oracle Database Reference*.

Defining and Controlling Database Links

The information here is specific to the gateway. For additional information on database links, refer to the *Oracle Database Reference*.

Link Accessibility

The database link should be accessible to a given user. A public database link can be used by any user ID. A private database link can be used only by the user who created it. The server makes no distinction regarding the type of use (such as read-only versus update or write) or accessibility of remote objects. The DRDA database, which is accessed, is responsible for these distinctions.

Links and CONNECT Clauses

The `CONNECT` clause is another security-related attribute of a database link. You can use the `CONNECT` clause to specify an explicit user ID and password, which can differ from the user's Oracle database user ID and password. This `CONNECT` user ID and password combination is sent to the gateway when the database link connection is first opened. Depending on gateway options, the gateway might send that user ID and password to the DRDA Server for validation.

If a database link is created without a `CONNECT` clause, then the user's Oracle database user ID and password are sent to the gateway when the connection is opened. If the user logs into the Oracle database with operating system authentication, then the gateway does not receive any user ID or password from the Oracle database. In this case, user ID mapping facilities at the DRDA Server can be used to make such a connection possible if all users on the same host can use the same DRDA database user ID.

Processing Inbound Connections

Current DRDA Servers provide options for manipulating the security conduct of an inbound (client) DRDA session request.

User ID Mapping

The most useful DRDA Server security capability is user ID mapping. User ID mapping refers to changing the user ID associated with an incoming DRDA request to some other user ID known to that server. This is a useful feature if your Oracle Database Gateway installation does not have a uniform user ID structure across all systems and databases.

DB2/OS390

The DB2 DDF Communication Database (CDB) stores inbound DRDA session security options.

These tables, pertinent to inbound sessions, have a role in security processing:

- `SYSIBM.IPNAMES` table

  The `SYSIBM.IPNAMES` table controls inbound security conducted for TCP/IP based sessions, affecting all DRDA connections from a particular host system. This table also controls whether inbound connection user IDs are subject to translation or mapping.

- `SYSIBM.SYSUSERNAMES` table

  When translation is used, rows in the `SYSIBM.SYSUSERNAMES` table specify translated user IDs by IP name and inbound user ID. Default entries that pertain to all IPs and to all inbound user IDs can be made in both tables. The mapping table can also be used simply to indicate which inbound user IDs are permitted from a particular IP or from all IPs, whether or not they are mapped.

This implementation provides a flexible mapping structure. You can specify that all connections from a particular IP use a single DB2 user ID, or that a particular inbound user ID always be mapped to a particular DB2 user ID regardless of origin. A `SYSUSERNAMES` entry with blank IP name and inbound user ID can designate a single default DB2 user ID for all connections unless a more specific entry, by IP name, user ID, or both, exists.

The CDB tables can be updated by a user with update privilege using a SQL tool such as the DB2 `SPUFI` utility. For example, most database administrators, systems programmers, and security officers can update CDB tables. The DB2 DDF component must be stopped and restarted for CDB changes to take effect.

The DB2 non-DRDA-specific security features are also involved in DRDA connections. User IDs are subject to normal DB2 or SAF/RACF validation in addition to connection or sign-on exit processing. Passwords are also subject to validation. After the connection is established, all normal authorizations or `GRANT`s associated with the user ID are in effect. The user ID must have execute privilege on the gateway DRDA package to process any SQL statements.

DB2/400

DB2/400 does not provide a user ID mapping capability comparable to that in DB2/OS390. Normally, the user ID in an incoming DRDA connection request must be a valid user ID on that DB2/400.

The DB2/400 subsystem communications entry for the gateway should specify that the gateway is not a secure location and should include a default user ID of `*NONE`.

After the application has completed the DRDA connection to the DB2/400, it is subject to all authorities and `GRANT`s associated with the user ID in use.

The user ID must have execute authority on the gateway DRDA package to execute any SQL statements.

DB2/Universal Database

DB2/Universal Database (DB2/UDB) does not provide a user ID mapping capability comparable to that in DB2/OS390. Normally, the user ID in an incoming DRDA connection request must be a valid user ID on the DB2/UDB host.

After the application has completed the DRDA connection to the DB2 host, it is subject to all authorities and `GRANTs` associated with the user ID in use. The user ID must have execute authority on the gateway DRDA package to execute any SQL statements.

Passwords in the Gateway Initialization File

The gateway uses user IDs and passwords to access the information in the remote database on the DRDA Server. Some user IDs 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. An encryption feature has been added as part of Heterogeneous Services' generic connectivity to help make this more secure. This feature is accessible by this gateway. With it Initialization parameters which contain sensitive values might be stored in an encrypted form. Refer to section 4.2.3, "Encrypting Initialization Parameters" in the manual *Oracle Database Heterogeneous Connectivity Administrator's Guide*, for information about how to use the feature.

Target reqd in 4.2.3 of HETER

See Also: the parameters `DRDA_RECOVERY_USERID` and `DRDA_RECOVERY_PASSWORD` in Appendix C, "Initialization Parameters" as examples, for more information.

16

Migration From Previous Releases

This chapter describes how to migrate to new instances of Oracle Database Gateway for DRDA from an existing installation. Perform the following steps to migrate to a new release of Oracle Database Gateway for DRDA from an existing release:

1. Install the New Release
2. Copy the Gateway Initialization Parameter File
3. Update the Initialization Parameters
4. Bind Gateway Package
5. Install/Upgrade Data Dictionary Views

Install the New Release

Install the new release of the gateway in a separate directory, as discussed in Chapter 12, "Installing Oracle Database Gateway for DRDA".

> **Caution:** Do not install the gateway over a previously existing gateway installation. This corrupts the existing installation.

Copy the Gateway Initialization Parameter File

Copy the `initsid.ora` from the old gateway instance to the new instance. The formats of the parameters in the `initsid.ora` Gateway Initialization File have changed. Refer to "Oracle Database Gateway for DRDA Initialization Parameters" on page C-6.

Update the Initialization Parameters

The next step in migrating to a new release of Oracle Database Gateway for DRDA consists of updating the initialization parameters.

Changed Parameters

The use of `DRDA_CONNECT_PARM` has changed in this version. Refer to Appendix C, "Initialization Parameters" for the syntax of the parameter.

Obsolete Parameters

The following parameters are obsolete for the 11g version. Remove them from your configuration files:

- MODE
- SERVER_PATH
- DRDA_OVERRIDE_FROM_CODEPAGE
- DRDA_OVERRIDE_TO_CODEPAGE
- ERROR_LOGGING
- ERROR_REPORTING
- ERRORTAG
- GATEWAY_SID
- GROUP_BY_OFF
- GTWDEBUG
- INCREMENT_CURSORS
- DRDA_CALLDESC_STMT
- DRDA_CALLDESC_PROC

Bind Gateway Package

When upgrading to 11g release you must rebind the gateway package if you have changed any of the following initialization parameters:

- DRDA_DISABLE_CALL
- DRDA_ISOLATION_LEVEL
- DRDA_PACKAGE_COLLID
- DRDA_PACKAGE_CONSTOKEN
- DRDA_PACKAGE_NAME
- DRDA_PACKAGE_OWNER
- DRDA_PACKAGE_SECTIONS

Install/Upgrade Data Dictionary Views

You must install or upgrade the data dictionary views when upgrading the Oracle Database Gateway for DRDA. Refer to Chapter 14, "Configuring Oracle Database Gateway for DRDA" for more information on creating data dictionary views.

Part VIII

Removing Oracle Database Gateway

Part VIII, "Removing Oracle Database Gateway" describes how to remove Oracle Database Gateways.

It contains the following chapter:

- Chapter 17, "Removing Oracle Database Gateway"

17

Removing Oracle Database Gateway

If you decide to remove the Oracle Database Gateway on Microsoft Windows (32-bit), then perform the following steps:

1. Insert the gateway product installation media.
2. Start the Oracle Universal Installer by running the Oracle Universal Installer executable, `setup.exe`:
    1. From the Start menu, select **Run**.
    2. Enter the path of the executable file name. For example:

        ```
        D: \Disk1\setup.exe
        ```

3. Step through the Oracle Universal Installer. Use the prompts listed in Table 17–1, " Steps to Remove the Gateway Using Oracle Universal Installer" as a guide for removing, following the instructions in the Response column.

Table 17–1 Steps to Remove the Gateway Using Oracle Universal Installer

Prompt	Response
1. Oracle Universal Installer: Welcome	Click **Deinstall Products**.
2. Inventory	You may either choose to remove *all* products that you may have installed at the time of your original gateway installation, or you may choose to deinstall only an instance of the gateway. ▪ To remove *all* products: Select the *ORACLE_HOME* where the gateway was installed. ▪ Click **Remove...** ▪ To remove *only* one instance of the gateway, expand the rows within the *ORACLE_HOME* recursively until you arrive at the folder. ▪ Click **Remove...**
3. Confirmation	Click **Yes**.
4. Inventory	Click **Close**.
5. Oracle Universal Installer: Welcome	Click **Cancel**.
6. Exit	Click **Yes**.

4. The the gateway is now removed.

When the Oracle Universal Installer confirms that the deinstallation has ended, verify that the removal procedure was successful. To do this, read the contents of the deinstallation log file, which is located in the `C:\Program Files\Oracle\Inventory\logs` directory.

The default file name is InstallActions*YYYY-MM-DD_HH-mm-SS-AM/PM*.log, where:

YYYY is year
MM is month
DD is day
HH is hour
mm is minute
SS is seconds
AM/PM is daytime or evening

These variables in the log file name represent the date and time the product was deinstalled.

5. The only files that are removed are those that were copied to the *`ORACLE_HOME`* directory during the gateway installation. You must remove any other related files manually, including deleting `listener.ora` and `tnsnames.ora` entries relating to the gateway, dropping database links.

Part IX

Appendixes

Part IX, "Appendixes" includes appendixes containing information relevant to installing and configuring Oracle Database Gateways.

It contains the following chapters:

- Appendix A, "Using Response Files for Noninteractive Installation"
- Appendix B, "Oracle Database Gateway Troubleshooting"
- Appendix D, "Configuration Worksheet for DRDA"
- Appendix C, "Initialization Parameters"
- Appendix E, "Globalization Support for DRDA"

A

Using Response Files for Noninteractive Installation

This appendix describes how to install and configure Oracle products using response files. It includes the following topics:

Introduction

Typically, Oracle Universal Installer runs in interactive mode, which means that it prompts you to provide information in graphical user interface (GUI) screens. Alternatively, you can run Oracle Universal Installer in noninteractive mode. Noninteractive mode is also referred to as silent mode, or silent installation.

You may want to use noninteractive mode to install Oracle Database Gateway on Microsoft Windows (32-bit) in the following scenarios:

- You need to deploy Oracle Components to multiple nodes in an unattended manner. You can schedule the noninteractive installation mode from the operating system scheduler or other job subsystem that your site normally uses.
- No interaction with the user is intended.
- A graphical facility to run Oracle Universal Installer in interactive mode is not available. (Oracle Universal Installer is always available on Windows, but not on UNIX systems.)

This section covers the following topics on how you can use response files to run Oracle Universal Installer in noninteractive mode:

Using Response Files to Install Oracle Components in Noninteractive Mode

To use noninteractive mode, you run Oracle Universal Installer with a response file. A response file is a text file that contains variables and values that Oracle Universal Installer uses during the installation process. Oracle provides a set of sample response files that you can customize, or you can create your own response file by recording your installation selections.

See Also: *Oracle Universal Installer and OPatch User's Guide* for more information about response file formats

Customizing a Sample Response File

Oracle provides response file templates for each product and installation type, and for each configuration tool. The response files for Oracle Gateway, tg.rsp and netca.rsp, are located in the response directory on the media.

Note: If you copied the software to a hard disk, then the response files are located in the `Disk1\response` directory.

To copy and modify a response file:

1. Copy the necessary response files from the \Response directory on the installation media to your hard drive.
2. From the Start menu, select **Programs**, then **Oracle - *HOME_NAME***, then **Oracle Installation Products**, then **Universal Installer Concepts Guide**.

   Oracle Universal Installer and OPatch User's Guide appears in HTML format.
3. Modify the response files with any text file editor by following the instructions in both the response files and *Oracle Universal Installer and OPatch User's Guide*.
4. Run the response file by following the instructions in the "Running Oracle Universal Installer and Specifying a Response File" section on page A-3.

Creating a New Response File

When you run Oracle Universal Installer in interactive mode, you can record your installation selections into a response file. You do this by running Oracle Universal Installer in Record mode. Oracle Universal Installer generates the response file immediately after you complete the **Summary** page, so you do not need to actually install the gateway to create the response file.

If you want to use the Record mode during a noninteractive installation, then Oracle Universal Installer records the variable values that were specified in the original source response file into the new response file.

Note: You cannot use Record mode to create a response file based on the Basic installation type.

To create a new response file:

1. Make sure that the computer on which you are creating the response file has met the requirements described in respective gateway installation chapters.
2. At the command prompt, use the `cd` command to change to the directory that contains the Oracle Universal Installer `setup.exe` executable.

   On the installation media, `setup.exe` is located on Disk 1. If you want to run Oracle Universal Installer from an existing gateway installation, then you can find `setup.exe` in *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\oui\bin`.
3. Enter the following command:

   ```
   setup -record -destinationFile response_file_name
   ```

Replace *response_file_name* with the complete path for the new response file. For example:

```
setup -record -destinationFile C:\response_files\install_oracle11g
```

4. After Oracle Universal Installer launches, enter the installation settings, which will be recorded into the response file.
5. When the Summary page appears, do one of the following:
   - Click **Install** to continue with the installation.
   - Click **Cancel** if you want to only create the response file but not continue with the installation. The installation will stop, but the settings you have entered will be recorded to the response file.

   Later, Oracle Universal Installer will save your new response file using the path and file name you specified on the command line.
6. If necessary, make any environment-specific changes to the response file for the computer on which you will run it.
7. Run the response file by following the instructions in the "Running Oracle Universal Installer and Specifying a Response File" section.

Running Oracle Universal Installer and Specifying a Response File

You run Oracle Universal Installer at the command line, specifying a response file. The Oracle Universal Installer executable, `setup.exe`, provides several options. For help information on the full set of these options, run `setup.exe` with the `-help` option, for example:

```
C:\ORACLE_BASE\ORACLE_HOME\oui\bin> setup.exe -help
```

To run Oracle Universal Installer and specify a response file:

1. Start a command prompt.
2. Go to the directory where Oracle Universal Installer is installed.
3. From the command line, run Oracle Universal Installer with the corrct response file. For example:

   ```
   C:\ORACLE_BASE\ORACLE_HOME\oui\bin> setup.exe [-silent]
   [-nowelcome] [-nowait] -responseFile filename
   ```

Where...	Description
filename	Identifies the full path of the response file
`-silent`	Runs Oracle Universal Installer in silent mode and suppresses the Welcome screen. If you use `-silent`, then `-nowelcome` is not necessary.
`-nowelcome`	Suppresses the Welcome screen that appears during installation
`-nowait`	Closes the console window when the silent installation completes

See Also:

- *Oracle Universal Installer and OPatch User's Guide* for more information about installation using response files

B

Oracle Database Gateway Troubleshooting

This appendix contains information about troubleshooting. It includes the following topics:

Verifying Requirements

Before you try any of the troubleshooting steps in this appendix, do the following:

- Check the system requirements section of respective gateway installation chapters to ensure that the system meets the requirements and that you have completed all the preinstallation tasks.
- Read the release notes for the product on your platform before installing it. The release notes are available on the Oracle software installation media. You can find the latest version of the release notes on the Oracle Technology Network Web site:

  `http://www.oracle.com/technology/documentation/index.html`

What to Do if an Installation Error Occurs

If you encounter an error during installation then:

- Do not exit Oracle Universal Installer.
- If you clicked **Next** after you entered incorrect information about one of the installation screens, then click **Back** to return to the screen and correct the information.
- If you encounter an error while Oracle Universal Installer is copying or linking files, see the "Reviewing the Log of an Installation Session" section on page B-2.
- If you encounter an error while a configuration assistant is running, see the "Troubleshooting Configuration Assistants" section on page B-2.
- If you cannot resolve the problem, then remove the failed installation by following the steps listed in the "Cleaning Up After a Failed Installation" section on page B-3.

Reviewing the Log of an Installation Session

During an installation, Oracle Universal Installer records all the actions that it performs, in a log file. If you encounter problems during the installation, then review the log file for information about possible causes of the problem.

```
SYSTEM_DRIVE:\Program Files\Oracle\Inventory\logs
```

Log file names take the form:

```
installActionsdate_time.log
```

For example, if the installation occurred at 9:00:56 A.M. on May 14, 2004, then the log file would be named:

```
installActions2004-05-14_09-00-56-am.log
```

Note: Do not delete or manually alter the `Inventory` directory or its contents. Doing so can prevent Oracle Universal Installer from locating products that you install on your system.

Troubleshooting Configuration Assistants

To troubleshoot an installation error that occurs when a configuration assistant is running:

- Review the installation log files listed in the "Reviewing the Log of an Installation Session" section on page B-2.
- Review the specific configuration assistant log file located in the *ORACLE_BASE**ORACLE_HOME*\`cfgtoollogs` directory. Try to fix the issue that caused the error.
- If you see the “Fatal Error. Reinstall” message, then look for the cause of the problem by reviewing the log files. Refer to the "Fatal Errors" section on page B-2 for further instructions.

Configuration Assistant Failure

Oracle configuration assistant failures are noted at the bottom of the installation screen. The configuration assistant interface displays additional information, if available. The configuration assistant execution status is stored in the `installActions`*`date_time`*`.log` file.

The execution status codes are listed in the following table:

Status	Result Code
Configuration assistant succeeded	0
Configuration assistant failed	1
Configuration assistant cancelled	-1

Fatal Errors

If you receive a fatal error while a configuration assistant is running then:

1. Remove the failed installation as described in the "Cleaning Up After a Failed Installation" section on page B-3.
2. Correct the cause of the fatal error.
3. Reinstall the Oracle software.

Noninteractive Installation Response File Error Handling

To determine whether a noninteractive installation succeeded or failed, check the `installActions`*`date_time`*`.log` file, located in *`SYSTEM_DRIVE`*`:\Program Files\Oracle\Inventory\logs`.

If necessary, then see the previous section for information about determining the location of the `Inventory` directory.

A silent installation fails if:

- You do not specify a response file.
- You specify an incorrect or incomplete response file.
- Oracle Universal Installer encounters an error, such as insufficient disk space.

Oracle Universal Installer or a configuration assistant validates the response file at run time. If the validation fails, then the noninteractive installation or configuration process ends. Oracle Universal Installer treats values for parameters that are of the wrong context, format, or type as if no value was specified in the file.

Cleaning Up After a Failed Installation

If an installation fails, you must remove files that Oracle Universal Installer created during the attempted installation and remove the Oracle home directory. Follow the instructions in Chapter 17, "Removing Oracle Database Gateway" to run Oracle Universal Installer to remove the gateway, manually remove the Oracle directory, and remove Oracle from the Registry Editor keys. Later, reinstall the software.

C

Initialization Parameters

The Oracle database initialization parameters in the `init.ora` file are distinct from gateway initialization parameters. Set the gateway parameters in the initialization parameter file using an agent-specific mechanism, or set them in the Oracle data dictionary using the `DBMS_HS` package. The gateway initialization parameter file must be available when the gateway is started.

This appendix contains a list of the gateway initialization parameters that can be set for each gateway and their description. It also describes the initialization parameter file syntax. It includes the following sections:

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. If there is a syntax error in an initialization parameter file, none of the settings take effect.
5. Set the parameter values as follows:

   ```
   [SET][PRIVATE] parameter=value
   ```

   Where:

   parameter is an initialization parameter name. It is a string of characters starting with a letter and consisting of letters, digits and underscores. Initialization parameter names are case sensitive.

value is the initialization parameter value. It is case-sensitive. 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.

For example, to enable tracing for an agent, set the HS_FDS_TRACE_LEVEL initialization parameter as follows:

```
HS_FDS_TRACE_LEVEL=ON
```

SET and PRIVATE are optional keywords. You cannot use either as an initialization parameter name. Most parameters are needed only as initialization parameters, so you usually do not need to use the SET or PRIVATE keywords. If you do not specify either SET or PRIVATE, the parameter is used only as an initialization parameter for the agent.

SET specifies that, in addition to being used as an initialization parameter, the parameter value is set as an environment variable for the agent process. Use SET for parameter values that the drivers or non-Oracle system need as environment variables.

PRIVATE specifies that the initialization parameter should be private to the agent and should not be uploaded to the Oracle database. Most initialization parameters should not be private. If, however, you are storing sensitive information 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 the initialization parameters private prevents the upload from happening and they do not appear in dynamic performance views. Use PRIVATE for the initialization parameters only if the parameter value includes sensitive information such as a username or password.

SET PRIVATE specifies that the parameter value is set as an environment variable for the agent process and is also private (not transferred to the Oracle database, not appearing in dynamic performance views or graphical user interfaces).

Oracle Database Gateway for Sybase Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for Sybase. They are as follows:

- HS_CALL_NAME
- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME

- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_LONG_PIECE_TRANSFER_SIZE
- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TIME_ZONE
- HS_TRANSACTION_MODEL
- IFILE
- HS_FDS_CONNECT_INFO
- HS_FDS_DEFAULT_OWNER
- HS_FDS_PROC_IS_FUNC
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_RESULTSET_SUPPORT
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_FETCH_ROWS

Oracle Database Gateway for Informix Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for Informix. They are as follows:

- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME
- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_LONG_PIECE_TRANSFER_SIZE
- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TIME_ZONE
- HS_TRANSACTION_MODEL
- IFILE
- HS_FDS_CONNECT_INFO
- HS_FDS_DEFAULT_OWNER
- HS_FDS_RECOVERY_ACCOUNT

- HS_FDS_RECOVERY_PWD
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_FETCH_ROWS

Oracle Database Gateway for Teradata Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for Teradata. They are as follows:

- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME
- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_LONG_PIECE_TRANSFER_SIZE
- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TIME_ZONE
- HS_TRANSACTION_MODEL
- IFILE
- HS_FDS_CONNECT_INFO
- HS_FDS_DEFAULT_OWNER
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_FETCH_ROWS

Oracle Database Gateway for SQL Server Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for Micorsoft SQL Server. They are as follows:

- HS_CALL_NAME
- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME
- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_LONG_PIECE_TRANSFER_SIZE

- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TIME_ZONE
- HS_TRANSACTION_MODEL
- IFILE
- HS_FDS_CONNECT_INFO
- HS_FDS_DEFAULT_OWNER
- HS_FDS_PROC_IS_FUNC
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_REPORT_REAL_AS_DOUBLE
- HS_FDS_RESULTSET_SUPPORT
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_FETCH_ROWS

Oracle Database Gateway for ODBC Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for ODBC. They are as follows:

- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME
- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_LONG_PIECE_TRANSFER_SIZE
- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TIME_ZONE
- IFILE
- HS_FDS_CONNECT_INFO
- HS_FDS_DEFAULT_OWNER
- HS_FDS_TRACE_LEVEL
- HS_TRANSACTION_MODEL

Oracle Database Gateway for DRDA Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for DRDA. They are as follows:

- HS_CALL_NAME
- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME
- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TRANSACTION_MODEL
- HS_FDS_FETCH_ROWS
- IFILE
- DRDA_CACHE_TABLE_DESC
- DRDA_CAPABILITY
- DRDA_CODEPAGE_MAP
- DRDA_COMM_BUFLEN
- DRDA_CONNECT_PARM
- DRDA_DEFAULT_CCSID
- DRDA_DESCRIBE_TABLE
- DRDA_DISABLE_CALL
- DRDA_FLUSH_CACHE
- DRDA_GRAPHIC_CHAR_SIZE
- DRDA_GRAPHIC_PAD_SIZE
- DRDA_GRAPHIC_LIT_CHECK
- DRDA_GRAPHIC_TO_MBCS
- DRDA_ISOLATION_LEVEL
- DRDA_LOCAL_NODE_NAME
- DRDA_MBCS_TO_GRAPHIC
- DRDA_OPTIMIZE_QUERY
- DRDA_PACKAGE_COLLID
- DRDA_PACKAGE_CONSTOKEN
- DRDA_PACKAGE_NAME
- DRDA_PACKAGE_OWNER
- DRDA_PACKAGE_SECTIONS

- DRDA_READ_ONLY
- DRDA_RECOVERY_PASSWORD
- DRDA_RECOVERY_USERID
- DRDA_REMOTE_DB_NAME
- FDS_CLASS
- HS_NLS_NCHAR
- LOG_DESTINATION
- ORA_MAX_DATE
- ORA_NLS11
- ORACLE_DRDA_TCTL
- ORACLE_DRDA_TRACE
- TRACE_LEVEL
- HS_NLS_DATE_FORMAT
- HS_NLS_DATE_LANGUAGE
- HS_NLS_NUMERIC_CHARACTER

Initialization Parameter Description

The following sections describe all the initialization file parameters that can be set for gateways.

HS_CALL_NAME

Property	Description
Default value	None
Range of values	Not applicable

Specifies the remote functions that can be referenced in SQL statements. The value is a list of remote functions and their owners, separated by semicolons, in the following format:

owner_name.function_name

For example:

```
owner1.A1;owner2.A2;owner3.A3
```

If an owner name is not specified for a remote function, the default owner name becomes the user name used to connect to the remote database (specified when the Heterogeneous Services database link is created or taken from user session if not specified in the DB link).

The entries for the owner names and the function names are case-sensitive.

HS_DB_DOMAIN

Property	Description
Default value	WORLD
Range of values	1 to 199 characters

Specifies a unique network sub-address for a non-Oracle system. The HS_DB_DOMAIN initialization parameter is similar to the DB_DOMAIN initialization parameter, described in the *Oracle Database Reference*. The HS_DB_DOMAIN initialization parameter is required if you use the Oracle Names server. The HS_DB_NAME and HS_DB_DOMAIN initialization parameters define the global name of the non-Oracle system.

Note: The HS_DB_NAME and HS_DB_DOMAIN initialization parameters must combine to form a unique address in a cooperative server environment.

HS_DB_INTERNAL_NAME

Property	Description
Default value	01010101
Range of values	1 to 16 hexadecimal characters

Specifies a unique hexadecimal number identifying the instance to which the Heterogeneous Services agent is connected. This parameter's value is used as part of a transaction ID when global name services are activated. Specifying a nonunique number can cause problems when two-phase commit recovery actions are necessary for a transaction.

HS_DB_NAME

Property	Description
Default value	HO
Range of values	1 to 8 characters

Specifies a unique alphanumeric name for the data store given to the non-Oracle system. This name identifies the non-Oracle system within the cooperative server environment. The HS_DB_NAME and HS_DB_DOMAIN initialization parameters define the global name of the non-Oracle system.

HS_DESCRIBE_CACHE_HWM

Property	Description
Default value	100
Range of values	1 to 4000

Specifies the maximum number of entries in the describe cache used by Heterogeneous Services. This limit is known as the describe cache high water mark. The cache contains descriptions of the mapped tables that Heterogeneous Services reuses so that it does not have to re-access the non-Oracle data store.

If you are accessing many mapped tables, increase the high water mark to improve performance. Increasing the high water mark improves performance at the cost of memory usage.

HS_LANGUAGE

Property	Description
Default value	System-specific
Range of values	Any valid language name (up to 255 characters)

Provides Heterogeneous Services with character set, language, and territory information of the non-Oracle data source. The value must use the following format:

language[*_territory.character_set*]

> **Note:** The globalization support initialization parameters affect error messages, the data for the SQL Service, and parameters in distributed external procedures.

Character Sets

Ideally, the character sets of the Oracle database server and the non-Oracle data source are the same. If they are not the same, Heterogeneous Services attempts to translate the character set of the non-Oracle data source to the Oracle database character set, and back again. The translation can degrade performance. In some cases, Heterogeneous Services cannot translate a character from one character set to another.

> **Note:** The specified character set must be a superset of the operating system character set on the platform where the agent is installed.

Language

The language component of the `HS_LANGUAGE` initialization parameter determines:

- Day and month names of dates
- AD, BC, PM, and AM symbols for date and time
- Default sorting mechanism

Note that Oracle does not determine the language for error messages for the generic Heterogeneous Services messages (`ORA-25000` through `ORA-28000`). These are controlled by the session settings in the Oracle database server.

> **Note:** Use the `HS_NLS_DATE_LANGUAGE` initialization parameter to set the day and month names, and the AD, BC, PM, and AM symbols for dates and time independently from the language.

Territory

The territory clause specifies the conventions for day and week numbering, default date format, decimal character and group separator, and ISO and local currency symbols. Note that the level of globalization support between the Oracle database server and the non-Oracle data source depends on how the gateway is implemented.

HS_LONG_PIECE_TRANSFER_SIZE

Property	Description
Default value	64 KB
Range of values	Any value up to 2 GB

Sets the size of the piece of `LONG` data being transferred. A smaller piece size means less memory requirement, but more round-trips to fetch all the data. A larger piece size means fewer round-trips, but more of a memory requirement to store the intermediate pieces internally. Thus, the initialization parameter can be used to tune a system for the best performance, with the best trade-off between round-trips and memory requirements, and network latency.

HS_OPEN_CURSORS

Property	Description
Default value	`50`
Range of values	`1` to the value of Oracle's `OPEN_CURSORS` initialization parameter

Defines the maximum number of cursors that can be open on one connection to a non-Oracle system instance.

The value never exceeds the number of open cursors in the Oracle database server. Therefore, setting the same value as the `OPEN_CURSORS` initialization parameter in the Oracle database server is recommended.

HS_RPC_FETCH_REBLOCKING

Property	Description
Default value	`ON`
Range of values	`OFF` or `ON`

Controls whether Heterogeneous Services attempts to optimize performance of data transfer between the Oracle database server and the Heterogeneous Services agent connected to the non-Oracle data store.

The following values are possible:

- `OFF` disables reblocking of fetched data so that data is immediately sent from agent to server.
- `ON` enables reblocking, which means that data fetched from the non-Oracle system is buffered in the agent and is not sent to the Oracle database server until the amount of fetched data is equal to or higher than the value of `HS_RPC_FETCH_`

SIZE initialization parameter. However, any buffered data is returned immediately when a fetch indicates that no more data exists or when the non-Oracle system reports an error.

HS_RPC_FETCH_SIZE

Property	Description
Default value	50000
Range of values	1 to 10000000

Tunes internal data buffering to optimize the data transfer rate between the server and the agent process.

Increasing the value can reduce the number of network round-trips needed to transfer a given amount of data, but also tends to increase data bandwidth and to reduce response time or *latency* as measured between issuing a query and completion of all fetches for the query. Nevertheless, increasing the fetch size can increase latency for the initial fetch results of a query, because the first fetch results are not transmitted until additional data is available.

HS_TIME_ZONE

Property	Description
Default value for '[+|-]hh:mm'	Derived from the NLS_TERRITORY initialization parameter
Range of values for '[+|-]hh:mm'	Any valid datetime format mask

Specifies the default local time zone displacement for the current SQL session. The format mask, [+|-]hh:mm, is specified to indicate the hours and minutes before or after UTC (Coordinated Universal Time—formerly Greenwich Mean Time). For example:

```
HS_TIME_ZONE = [+ | -] hh:mm
```

HS_TRANSACTION_MODEL

Property	Description
Default Value	COMMIT_CONFIRM
Range of Values	COMMIT_CONFIRM, READ_ONLY, SINGLE_SITE

Specifies the type of transaction model that is used when the non-Oracle database is updated by a transaction.

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the non-Oracle 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 non-Oracle 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.

    COMMIT_CONFIRM does not apply to Oracle Database Gateway for ODBC. The default value for Oracle Database Gateway for ODBC is SINGLE_SITE.

- READ_ONLY provides read access to the non-Oracle database.
- SINGLE_SITE provides read and write access to the non-Oracle database. However, the gateway cannot participate in distributed updates.

IFILE

Property	Description
Default value	None
Range of values	Valid parameter file names

Use the IFILE initialization parameter to embed another initialization file within the current initialization file. The value should be an absolute path and should not contain environment variables. The three levels of nesting limit does not apply.

See Also: *Oracle Database Reference*

HS_FDS_CONNECT_INFO

Property	Description
Default Value	None
Range of Values	Not applicable

HS_FDS_CONNECT_INFO which describes the connection to the non-Oracle system.

The default initialization parameter file already has an entry for this parameter. The syntax for HS_FDS_CONNECT_INFO for the gateways are as follows:

For Oracle Database Gateway for Sybase:

HS_FDS_CONNECT_INFO=*host_name*:*port_number*/*database_name*

where, *host_name* is the host name or IP address of the machine hosting the Sybase database, *port_number* is the port number of the Sybase database server, and *database_name* is the Sybase database name.

For Oracle Database Gateway for Informix:

HS_FDS_CONNECT_INFO=*host_name*:*port_number*/*server_name*/*database_name*

where, *host_name* is the host name or IP address of the machine hosting the Informix database, *port_number* is the port number of the Informix database server, *server_*

name is the name of the server machine for the Informix data, and *database_name* is the Informix database name.

For Oracle Database Gateway for Teradata:

```
HS_FDS_CONNECT_INFO=host_alias:port_number[/database_name]
```

where, *host_alias* is the host alias name or IP address of the machine hosting the Teradata database, *port_number* is the port number of the Teradata database server, and *database_name* is the Teradata database name. The *database_name* variable is optional.

For Oracle Database Gateway for Microsoft SQL Server:

```
HS_FDS_CONNECT_INFO= host_name/[instance_name][/database_name]
```

where, *host_name* is the host name or IP address of the machine hosting the SQL Server database, *instance_name* is the instance of SQL Server running on the machine, and *database_name* is the Microsoft SQL Server database name. Both *instance_name* and *database_name* are optional. If *instance_name* is omitted and *database_name* is provided, the slash (/) is required. This can be shown as follows:

```
HS_FDS_CONNECT_INFO= host_name//database_name
```

For Oracle Database Gateway for ODBC:

```
HS_FDS_CONNECT_INFO=dsn_value
```

where *dsn_value* is the name of the system DSN defined in the Windows ODBC Data Source Administrator.

HS_FDS_DEFAULT_OWNER

Property	Description
Default Value	None
Range of Values	Not applicable

The name of the table owner that is used for the non-Oracle database tables if an owner is not specified in the SQL statements.

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

HS_FDS_PROC_IS_FUNC

Property	Description
Default Value	FALSE
Range of Values	TRUE, FALSE

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

> **Note:** If you set this initialization parameter, you must change the syntax of the procedure execute statement for all existing stored procedures to handle return values.

HS_FDS_RECOVERY_ACCOUNT

Property	Description
Default Value	RECOVER
Range of values	Any valid user ID

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 the non-Oracle system. For more information about the commit-confirm model, see the `HS_TRANSACTION_MODEL` parameter.

The name of the recovery account is case-sensitive.

HS_FDS_RECOVERY_PWD

Property	Description
Default Value	`RECOVER`
Range of values	Any valid password

Specifies the password of the recovery account used for the commit-confirm transaction model set up at the non-Oracle system. For more information about the commit-confirm model, see the `HS_TRANSACTION_MODEL` parameter.

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

HS_FDS_RESULTSET_SUPPORT

Property	Description
Default Value	`FALSE`
Range of Values	`TRUE`, `FALSE`

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

> **Note:** If you set this initialization parameter, you must do the following:
>
> - Change the syntax of the procedure execute statement for all existing stored procedures, to handle result sets
> - Work in the sequential mode of Heterogeneous Services

HS_FDS_TRACE_LEVEL

Property	Description
Default Value	OFF
Range of values	OFF, ON, DEBUG

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 by default to a gateway log file in LOG directory where the gateway is installed.
- DEBUG enables the tracing of detailed error messages that can be used for debugging.

HS_FDS_TRANSACTION_LOG

Property	Description
Default Value	HS_TRANSACTION_LOG
Range of Values	Any valid table name

Specifies the name of the table created in the non-Oracle system for logging transactions. For more information about the transaction model, see the HS_TRANSACTION_MODEL parameter.

HS_FDS_REPORT_REAL_AS_DOUBLE

Property	Description
Default Value	FALSE
Range of Values	TRUE, FALSE

Enables Oracle Database Gateway for Microsoft SQL Server to treat SINGLE FLOAT PRECISION fields as DOUBLE FLOAT PRECISION fields.

HS_FDS_FETCH_ROWS

Property	Description
Default Value	100
Range of Values	Any integer between 1 and 1000
Syntax	HS_FDS_FETCH_ROWS=*num*

HS_FDS_FETCH_ROWS specifies the fetch array size. This is the number of rows to be fetched from the non-Oracle database and to return to Oracle database at one time.

This parameter will be affected by the HS_RPC_FETCH_SIZE and HS_RPC_FETCH_REBLOCKING parameters.

DRDA_CACHE_TABLE_DESC

Property	Description
Default Value	TRUE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_CACHE_TABLE_DESC=*{TRUE|FALSE}*

DRDA_CACHE_TABLE_DESC directs the gateway to cache table descriptions once per transaction. This can reduce the number of table lookups requested by Oracle database and can speed up the execution of SQL statements. You may wish to disable this option if you would be altering the structure of a remote table and if you would be examining it within the same transaction.

DRDA_CAPABILITY

Property	Description
Default Value	None
Range of Values	Refer to Chapter 4, "Developing Applications" in *Oracle Database Gateway for DRDA User's Guide*
Syntax	DRDA_CAPABILITY={*FUNCTION/{ON|OFF}*},...

DRDA_CAPABILITY specifies which mapped functions of Oracle database will be treated natively. In other words, no special pre processing or post processing will be done for these functions. They will be passed through to the DRDA Server unmodified.

DRDA_CODEPAGE_MAP

Property	Description
Default Value	codepage.map
Range of Values	Any valid file path
Syntax	DRDA_CODEPAGE_MAP=*codepage.map*

DRDA_CODEPAGE_MAP specifies the location of the codepage map. You may specify only the filename, which will be searched for within the $ORACLE_HOME/tg4drda/admin directory, or you may specify the full path name of the file.

DRDA_COMM_BUFLEN

Property	Description
Default Value	32767

Property	Description
Range of Values	512 through 32767
Syntax	DRDA_COMM_BUFLEN=*num*

DRDA_COMM_BUFLEN specifies the communications buffer length. This is a number indicating the TCP/IP buffer size in bytes.

DRDA_CONNECT_PARM

Property	Description
Default Value	DRDACON1:446
Range of Values	Any alphanumeric string 1 to 255 characters in length
Syntax	DRDA_CONNECT_PARM={*hostname*|*ip_address*}{*:port*}

DRDA_CONNECT_PARM specifies the TCP/IP hostname or IP Address of the DRDA Server and, as an option, the Service Port number on which the DRDA Server is listening.

The DRDA standard specifies that port 446 be used for DRDA services. However, if several DRDA servers are operating on the same system, then they will need to provide service on different ports. Therefore, the port number that is used by each DRDA server will need to be extracted from the configuration of each individual DRDA server. DB2 for OS/390 and DB2/400 typically use the DRDA standard port number, 446, whereas DB2/UDB typically uses 50000 as the port number. Refer to IBM DB2 Administrator and Installation guides for locating and changing these port numbers for your DRDA server. For additional information, consult your DB2 DBA or System Administrator.

DRDA_DEFAULT_CCSID

Property	Description
Default Value	None
Range of Values	Any supported DRDA Server CCSID
Syntax	DRDA_DEFAULT_CCSID=*ccsid*

DRDA_DEFAULT_CCSID specifies the default CCSID or character set codepage for character set conversions when the DRDA Server database indicates that a character string has a CCSID of 65535. DRDA Servers use CCSID 65535 for columns specified as "FOR BIT DATA". In most cases, this parameter should not be specified, allowing CCSID 65535 to be treated as an Oracle RAW data type.

This parameter is for supporting databases (in particular, DB2/400) that use CCSID 65535 as the default for all tables created. Allowing CCSID 65535 to be treated as another CCSID can save such sites from having to modify every table.

Warning: **Specifying any value for DRDA_DEFAULT_CCSID causes all "FOR BIT DATA" columns to be handled as text columns that need character set conversion and, therefore, any binary data in these columns can encounter conversion errors (ORA-28527).**

DRDA_DESCRIBE_TABLE

Property	Description
Default Value	TRUE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_DESCRIBE_TABLE={*TRUE*|*FALSE*}

DRDA_DESCRIBE_TABLE directs the gateway to use the DRDA operation Table Describe to return the description of tables. This is an optimization that reduces the amount of time and resources that are used to look up the definition of a table.

Note: This feature is not compatible with DB2 Aliases or Synonyms. If you use DB2 aliases, then be sure to disable this option.

DRDA_DISABLE_CALL

Property	Description
Default Value	TRUE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_DISABLE_CALL={*TRUE*|*FALSE*}

DRDA_DISABLE_CALL controls stored procedure usage, and is also used to control how the package is bound on the target database. This parameter should be set to FALSE only for supported target DRDA servers and should be set to TRUE otherwise.

Note: Any change to this parameter requires you to rebind.

DRDA_FLUSH_CACHE

Property	Description
Default Value	SESSION
Range of Values	{SESSION|COMMIT}
Syntax	DRDA_FLUSH_CACHE={*SESSION*|*COMMIT*}

DRDA_FLUSH_CACHE specifies when the cursor cache is to be flushed. With DRDA_FLUSH_CACHE=COMMIT, the cursor cache is flushed whenever the transaction is

committed. With DRDA_FLUSH_CACHE=SESSION, the cache is not flushed until the session terminates.

DRDA_GRAPHIC_CHAR_SIZE

Property	Description
Default Value	4
Range of Values	1 through 4
Syntax	DRDA_GRAPHIC_CHAR_SIZE=num

DRDA_GRAPHIC_CHAR_SIZE is used to define the character conversion size to be used for GRAPHIC data types. It is a tuning parameter which affects the maximum size of a GRAPHIC data type when the column is described.

DRDA_GRAPHIC_PAD_SIZE

Property	Description
Default Value	0
Range of Values	0 through 127
Syntax	DRDA_GRAPHIC_PAD_SIZE=num

DRDA_GRAPHIC_PAD_SIZE is used to pad the size of a Graphic column as described by the DRDA Server. This is sometimes necessary depending upon the character set of the DRDA database and Oracle database. If the Oracle database is based on EBCDIC and the DRDA database is based on ASCII, then a pad size of 2 may be needed.

DRDA_GRAPHIC_LIT_CHECK

Property	Description
Default Value	FALSE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_GRAPHIC_LIT_CHECK={*TRUE*|*FALSE*}

DRDA_GRAPHIC_LIT_CHECK directs the gateway to evaluate string literals within INSERT SQL statements in order to determine if they need to be converted to double-byte format for insertion into a Graphic column at the DRDA Server database. This is done by querying the column attributes of the table in the SQL statement to determine if a string literal is being applied to a column with a Graphic data type. If the table column is Graphic, and if this parameter is TRUE, then the gateway will rewrite the SQL statement with the literal converted to double-byte format. Existing double-byte characters in the string will be preserved, and all single-byte characters will be converted to double-byte characters.

DRDA_GRAPHIC_TO_MBCS

Property	Description
Default Value	FALSE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_GRAPHIC_TO_MBCS={*TRUE*|*FALSE*}

DRDA_GRAPHIC_TO_MBCS directs the gateway to convert graphic data that has been fetched from the DRDA Server into Oracle multi-byte data, translating double-byte characters into single-byte characters where possible.

DRDA_ISOLATION_LEVEL

Property	Description
Default Value	CHG for DB2/400, CS for DB2/OS390, DB2/UDB
Range of Values	{CHG|CS|RR|ALL|NC}
Syntax	DRDA_ISOLATION_LEVEL={*CHG*|*CS*|*RR*|*ALL*|*NC*}

DRDA_ISOLATION_LEVEL specifies the isolation level that is defined to the package when it is created. All SQL statements that are sent to the remote DRDA database are executed with this isolation level. Isolation level seriously affects performance of applications. Use caution when specifying an isolation level other than the default. For information on isolation levels, refer to your IBM database manuals.

The following table lists the isolation levels and their descriptions. The levels are specified in ascending order of control, with CHG having the least reliable cursor stability and RR having the most. Note that higher stability uses more resources on the server and can lock those resources for extended periods.

Table C–1 Isolation Levels and Their Descriptions

Level	Description
CHG	Change (default for DB2/400)
CS	Cursor Stability (default for DB2/UDB, and DB2/OS390)
RR	Repeatable Read
ALL	ALL
NC	No Commit

Note: Any change to this parameter requires you to rebind.

DRDA_LOCAL_NODE_NAME

Property	Description
Default Value	AIX_RS6K
Range of Values	any alphanumeric string 1 to 8 characters in length

Property	Description
Syntax	DRDA_LOCAL_NODE_NAME=*name*

DRDA_LOCAL_NODE_NAME specifies the name by which the gateway will be known to the DRDA Server. This name is used internally by the DRDA Server to identify the local node.

DRDA_MBCS_TO_GRAPHIC

Property	Description
Default Value	FALSE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_MBCS_TO_GRAPHIC={*TRUE*|*FALSE*}

DRDA_MBCS_TO_GRAPHIC directs the gateway to convert multi-byte data (that has been sent from Oracle to the DRDA database) into pure double-byte characters. This parameter is primarily intended to be used with bind variables in order to ensure that the data is properly formatted and will therefore be acceptable to the DRDA Server. It applies only to INSERT SQL statements that are using bind variables. When used in combination with the DRDA_GRAPHIC_LIT_CHECK parameter, this parameter can help ensure that data that is being inserted into a Graphic column is handled correctly by the target DRDA Server.

DRDA_OPTIMIZE_QUERY

Property	Description
Default Value	TRUE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_OPTIMIZE_QUERY={*TRUE*|*FALSE*}

DRDA_OPTIMIZE_QUERY enables or disables the distributed query optimizer (DQO) capability. The DQO capability is useful for optimizing queries that access large amount of data, but it can add overhead to small queries.

> **See Also:** *Oracle Database gateway for DRDA User's Guide* for more information.

This parameter is valid only if the DRDA Server is DB2/OS390. If the DRDA Server is DB2/400 or DB2/UDB, then you must set the value to FALSE.

DRDA_PACKAGE_COLLID

Property	Description
Default Value	ORACLE
Range of Values	An alphanumeric string 1 to 18 characters in length
Syntax	DRDA_PACKAGE_COLLID=*collection_id*

DRDA_PACKAGE_COLLID specifies the package collection ID. Note that in DB2/400, the collection ID is actually the name of an AS/400 library.

> **Note:** Any change to this parameter requires you to rebind the package.

DRDA_PACKAGE_CONSTOKEN

Property	Description
Default Value	None, use the sample provided
Range of Values	A 16-digit hexadecimal number
Syntax	DRDA_PACKAGE_CONSTOKEN=*hexnum*

DRDA_PACKAGE_CONSTOKEN specifies the package consistency token. This is a 16-digit hexadecimal representation of an 8-byte token. Oracle recommends that you do not change the consistency token. The consistency token used at runtime must match the one used when the package is bound. The value depends on the DRDA Server being used.

> **Note:** Any change to this parameter requires you to rebind the package.

DRDA_PACKAGE_NAME

Property	Description
Default Value	G2DRSQL
Range of Values	An alphanumeric string 1 to 18 characters in length
Syntax	DRDA_PACKAGE_NAME=*name*

DRDA_PACKAGE_NAME specifies the package name. Note that the package is stored in the DRDA Server under this name as a SQL resource. Refer to the DRDA Server documentation for length limitations package names. Many typical implementations restrict the length to 8 characters.

> **Note:** Any change to this parameter requires that you rebind the package.

DRDA_PACKAGE_OWNER

Property	Description
Default Value	None
Range of Values	Any valid user ID
Syntax	DRDA_PACKAGE_OWNER=*userid*

DRDA_PACKAGE_OWNER specifies the database user ID that owns the package. This enables the owner to be a user other than the connected user ID when the package is created. The package owner must be the same user as the owner of the ORACLE2PC table.

Note: Any change to this parameter requires you to rebind the package.

DRDA_PACKAGE_SECTIONS

Property	Description
Default Value	100
Range of Values	Any integer between 1 and 65535
Syntax	DRDA_PACKAGE_SECTIONS=*num*

DRDA_PACKAGE_SECTIONS specifies the number of cursors declared at the remote database when the package is bound. This is the maximum number of open cursors permitted at any one time. Change this parameter only if an application needs more than 100 open concurrent cursors.

Note: Any change to this parameter requires you to rebind the package.

DRDA_READ_ONLY

Property	Description
Default Value	FALSE
Range of Values	{TRUE|FALSE}
Syntax	DRDA_READ_ONLY={*TRUE*|*FALSE*}

DRDA_READ_ONLY specifies whether the gateway runs in a read-only transaction mode. In this mode, SQL statements that modify data are not permitted.

DRDA_RECOVERY_PASSWORD

Property	Description
Default Value	none
Range of Values	any valid password
Syntax	DRDA_RECOVERY_PASSWORD=*passwd*

DRDA_RECOVERY_PASSWORD is used with the DRDA_RECOVERY_USERID. The recovery user connects to the IBM database if a distributed transaction is in doubt.

See Also: *Oracle Database gateway for DRDA User's Guide* for more information.

DRDA_RECOVERY_USERID

Property	Description
Default Value	ORARECOV
Range of Values	Any valid user ID
Syntax	DRDA_RECOVERY_USERID=*userid*

DRDA_RECOVERY_USERID specifies the user ID that is used by the gateway if a distributed transaction becomes in doubt. This user ID must have execute privileges on the package and must be defined to the IBM database.

If a distributed transaction becomes in doubt, then the Oracle database determines the status of the transaction by connecting to the IBM database, using the DRDA_RECOVERY_USERID. If this parameter is missing, then the gateway attempts to connect to a user ID of ORARECOV.

DRDA_REMOTE_DB_NAME

Property	Description
Default Value	DB2V2R3
Range of Values	An alphanumeric string 1 to 18 characters in length
Syntax	DRDA_REMOTE_DB_NAME=*name*

DRDA_REMOTE_DB_NAME specifies the DRDA Server location name. This is an identifying name that is assigned to the server for DRDA purposes. A technique for determining this name by using a SQL SELECT statement is discussed in each of the server-specific installation sections in Chapter 13, "Configuring the DRDA Server".

FDS_CLASS

Property	Description
Default Value	TG4DRDA_DB2MVS
Range of Values	Refer to the list below for valid values
Syntax	FDS_CLASS=*TG4DRDA_DB2MVS*

FDS_CLASS specifies the capability classification used by Oracle database and the gateway. These values may change from release to release, depending on whether the gateway capabilities change.

The valid default values for FDS_CLASS are as follows:

For a DB2/OS390 database: TG4DRDA_DB2MVS

For a DB2/400 database: TG4DRDA_DB2400

For a DB2/UDB database: TG4DRDA_DB2UDB

HS_NLS_NCHAR

Property	Description
Default Value	None
Range of Values	Any valid character set specification
Syntax	HS_NLS_NCHAR=*character_set*

HS_NLS_NCHAR specifies the character set that the gateway will use to interact with the DRDA Server when accessing Graphic data. Set this parameter to the same value as the character set component of the HS_LANGUAGE parameter. For additional details, refer to Appendix E, "Globalization Support for DRDA" and to the *Oracle Database Heterogeneous Connectivity Administrator's Guide*.

LOG_DESTINATION

Property	Description
Default Value	$ORACLE_HOME/tg4drda/log/gateway sid_ pid.log
Range of Values	Any valid file path
Syntax	LOG_DESTINATION=*logpath*

LOG_DESTINATION specifies the destination for gateway logging and tracing. This parameter should specify a file. If the file already exists, it will be overwritten.

After any failure to open the logpath, a second attempt to open the default is made.

Usually, LOG_DESTINATION should specify a directory. If it is specified as a file, and if two or more users simultaneously use the same instance of the gateway, then they are writing to the same log. The integrity of this log is not guaranteed. If you do not specify this parameter, then the default is assumed.

ORA_MAX_DATE

Property	Description
Default Value	4712-12-31
Range of Values	Any valid date less than 4712-12-31
Syntax	ORA_MAX_DATE=*yyyy-mm-dd*

ORA_MAX_DATE specifies the gateway maximum date value. If the fetched date value is larger than 4712-12-31, the gateway replaces the date value with the value defined by the ORA_MAX_DATE parameter. Any date between January 1, 4712 BC and December 31, 4712 AD is valid.

ORA_NLS11

Property	Description
Default Value	*ORACLE_HOME*\nls\data

Property	Description
Range of Values	Any valid Globalization Support directory path
Syntax	SET ORA_NLS11=*nlspath*

ORA_NLS11 specifies the directory to which the gateway loads its character sets and other language data. Normally this parameter does not need to be set. Some configurations, however, may require that it be set.

ORACLE_DRDA_TCTL

Property	Description
Default Value	None
Range of Values	Any valid file path
Syntax	ORACLE_DRDA_TCTL=*tracecontrolpath*

ORACLE_DRDA_TCTL specifies the path to the DRDA internal trace control file. This file contains module tracing commands. A sample file is stored in *ORACLE_HOME*\tg4drda\admin\debug.tctl. This parameter is used for diagnostic purposes.

ORACLE_DRDA_TRACE

Property	Description
Default Value	value specified for LOG_DESTINATION
Range of Values	any valid file path
Syntax	ORACLE_DRDA_TRACE=*logpath*

ORACLE_DRDA_TRACE is used to specify a different log path for DRDA internal tracing. This tracing is separate from the rest of the gateway tracing, as specified by the LOG_DESTINATION parameter. By default, this parameter will append the DRDA internal trace to the gateway trace. This parameter is used for diagnostic purposes.

TRACE_LEVEL

Property	Description
Default Value	0
Range of Values	0-255
Syntax	TRACE_LEVEL=*number*

TRACE_LEVEL specifies a code tracing level. This value determines the level of detail which is logged to the gateway logfile during execution. This parameter is primarily used for diagnostics.

HS_NLS_DATE_FORMAT

Property	Description
Default value	Value determined by the HS_LANGUAGE initialization parameter
Range of values	Any valid date format mask (up to 255 characters)

Defines the date format for dates used by the target system. This initialization parameter has the same function as the NLS_DATE_FORMAT initialization parameter for an Oracle database server. The value can be any valid date mask listed in the *Oracle Database SQL Language Reference*, but must match the date format of the target system. For example, if the target system stores the date February 14, 2001 as 2001/02/14, set the parameter to yyyy/mm/dd. Note that characters must be lowercase.

HS_NLS_DATE_LANGUAGE

Property	Description
Default value	Value determined by the HS_LANGUAGE initialization parameter
Range of values	Any valid NLS_LANGUAGE value (up to 255 characters)

Specifies the language used in character date values coming from the non-Oracle system. Date formats can be language independent. For example, if the format is dd/mm/yyyy, all three components of the character date are numeric. In the format dd-mon-yyyy, however, the month component is the name abbreviated to three characters. The abbreviation is language dependent. For example, the abbreviation for the month April is "apr", which in French is "avr" (Avril).

Heterogeneous Services assumes that character date values fetched from the non-Oracle system are in this format. Also, Heterogeneous Services sends character date bind values in this format to the non-Oracle system.

HS_NLS_NUMERIC_CHARACTER

Property	Description
Default value	Value determined by the HS_LANGUAGE initialization parameter
Range of values	Any valid NLS_NUMERIC_CHARACTERS value (any two valid numeric characters)

Specifies the characters to use as the group separator and the decimal character. The group separator separates integer groups (such as thousands, millions, and billions). The decimal character separates the integer portion of a number from the decimal portion.

D

Configuration Worksheet for DRDA

The table below is a worksheet that lists all of the parameter names and the reasons that you will need them for configuring the gateway and TCP/IP. Use the worksheet to gather the specific information that you need before you begin the configuration process.

Table D–1 List of Parameters Needed to Configure the Gateway

Reason	Name of Parameter Needed	Your Specific Parameters Here
Oracle home of the gateway	■ ORACLE_HOME	
System ID of the gateway	■ ORACLE_SID	
Configuring TCP/IP	■ Local Host name, Domain Name ■ IP Address ■ Network Mask ■ Name Server IP Address ■ DRDA server Host name or IP Address ■ DRDA server Service Port Number	
Recovery user ID	■ DRDA_RECOVERY_USERID	
Recovery Password	■ DRDA_RECOVERY_PASSWORD	
Remote Database Name	■ DRDA_REMOTE_DB_NAME	
Connection Parameter	■ DRDA_CONNECT_PARM	
Remote collection ID	DRDA_PACKAGE_COLLID	
Remote package name	DRDA_PACKAGE_NAME	
Owner ID of DRDA package	DRDA_PACKAGE_OWNER	
DB Name used with Oracle database	HS_DB_NAME	
DB Domain used with Oracle database	HS_DB_DOMAIN	

Note: The user ID that is used to bind or rebind the DRDA package must have the following privileges on the remote database; your database administrator will need to provide these.

- package privileges of BIND, COPY, and EXECUTE
- collection privilege of CREATE IN
- system privileges of BINDADD and BINDAGENT

E

Globalization Support for DRDA

This appendix documents the Globalization Support information for the Oracle Database Gateway for DRDA. This supplements the general Oracle Globalization Support information found in the *Oracle Database Advanced Application Developer's Guide*.

Globalization Support enables users to interact with Oracle applications in their native language, using their conventions for displaying data. The Oracle Globalization Support architecture is data-driven, enabling support for specific languages and character encoding schemes to be added without any changes in source code.

There are a number of different settings in the gateway, DRDA Server, Oracle Database 11*g* , and client that affect Globalization Support processing. In order for translations to take place correctly, character settings of these components must be compatible.

This appendix contains the following sections:

Overview of Globalization Support Interactions

Figure E–1 illustrates Globalization Support interactions within your system, including each component of your system and the parameters of each component that affect Globalization Support processing in a distributed environment. Table E–1 describes the architecture illustrated in Figure E–1.

Figure E–1 Architecture of Globalization Support Interactions with Your System Components

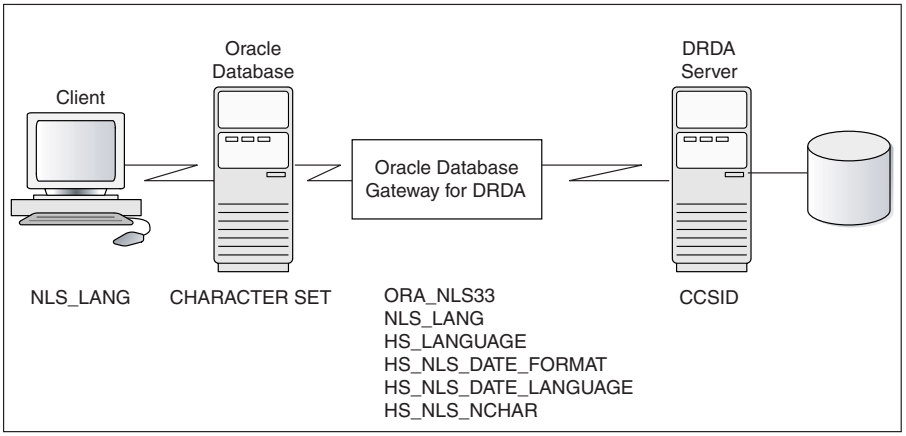


Table E–1 describes in detail the parameters and variables needed for Globalization Support processing within each of your system environments: the client environment, the Oracle database, the gateway, and the DRDA Server.

Table E–1 Parameters Needed for Globalization Support Processing in Your System Environments

Environment	Parameter or Variable	Description
Client	NLS_LANG	An environmental variable. NLS_LANG sets the Globalization Support environment that is used by the database, both for the server session and for the client application. This ensures that the language environments of both database and client application are automatically the same. Because NLS_LANG is an environment variable, it is read by the client applications at startup time. The client communicates the information defined in NLS_LANG to the server when it connects. Refer to "Client and Oracle Database Configuration" on page E-3 for detailed information.
Oracle database	CHARACTER SET	This option is set during creation of the database. CHARACTER SET determines the character encoding scheme that is used by the database. CHARACTER SET is defined at database creation in the CREATE DATABASE statement. All data columns of type CHAR, VARCHAR2, and LONG have their data stored in the database character set. Refer to "Client and Oracle Database Configuration" on page E-3 for detailed information.
Oracle Database Gateway for DRDA	ORA_NLS11	An environmental variable. ORA_NLS11 determines where the gateway loads its character sets and other language data. Refer to "Gateway Language Interaction with DRDA Server" on page E-4 for detailed information.
Oracle Database Gateway for DRDA	NLS_LANG	An environmental variable. NLS_LANG defines the character set that is used for communication between the gateway and the Oracle database. Refer to "Gateway Language Interaction with DRDA Server" on page E-4 for detailed information.
Oracle Database Gateway for DRDA	HS_LANGUAGE	An initialization parameter HS_LANGUAGE defines the character set that is used for communication between the gateway and the DRDA Server. Refer to "Gateway Language Interaction with DRDA Server" on page E-4 for detailed information.

Table E–1 (Cont.) Parameters Needed for Globalization Support Processing in Your System Environments

Environment	Parameter or Variable	Description
Oracle Database Gateway for DRDA	HS_NLS_NCHAR	An initialization parameter. HS_NLS_NCHAR defines the NCHAR character set that is used for communications between the gateway and the DRDA Server. This parameter is required when the gateway will be accessing GRAPHIC or multibyte data on the DRDA Server. Set this parameter to the same value as the character set component of the HS_LANGUAGE parameter. For detailed information, refer to "Gateway Language Interaction with DRDA Server" on page E-4.
Oracle Database Gateway for DRDA	HS_NLS_DATE_FORMAT	An initialization parameter. HS_NLS_DATE_FORMAT specifies the format for dates that are used by the DRDA Server. Refer to "Gateway Language Interaction with DRDA Server" on page E-4 for detailed information.
Oracle Database Gateway for DRDA	HS_NLS_DATE_LANGUAGE	An initialization parameter. HS_NLS_DATE_LANGUAGE specifies the language that is used by the DRDA Server for day and month names, and for date abbreviations. Refer to "Gateway Language Interaction with DRDA Server" on page E-4 for detailed information.
DRDA Server	CCSID	CCSID is the server character set that is mapped in the gateway to the equivalent Oracle character set. The CCSID specifies the character set that the DRDA database uses to store data. It is defined when you create your database. Refer to "Gateway Codepage Map Facility" on page E-5.

Client and Oracle Database Configuration

A number of Globalization Support parameters control Globalization Support processing between the Oracle database and client. You can set language-dependent action defaults for the server, and you can set language-dependent action for the client that overrides these defaults. For a complete description of Globalization Support parameters, refer to the Globalization Support chapter in the *Oracle Database Administrator's Guide*. These parameters do not directly affect gateway processing. However, you must ensure that the client character set (which is specified by the Oracle database NLS_LANG environment variable) is compatible with the character sets that you specify on the gateway and on the DRDA Server.

When you create the Oracle database, the character set that is used to store data is specified by the CHARACTER SET clause of the CREATE DATABASE statement. After the database is created, the database character set cannot be changed unless you re-create the database.

Normally, the default for CHARACTER SET is US7ASCII, which supports only the 26 Latin alphabetic characters. If you have specified 8-bit character sets on the gateway and DRDA Server, then you must have a compatible 8-bit character set defined on your database. To check the character set of an existing database, run the command:

```
SELECT USERENV('LANGUAGE') FROM DUAL;
```

For more information, refer to "Specifying Character Sets" in the *Oracle Database Administrator's Guide*.

Note that this does not mean that the gateway character set must be the same as the Oracle database character set. The Oracle Net facility will be performing implicit conversion between the Oracle database character set and the gateway character set.

Gateway Language Interaction with DRDA Server

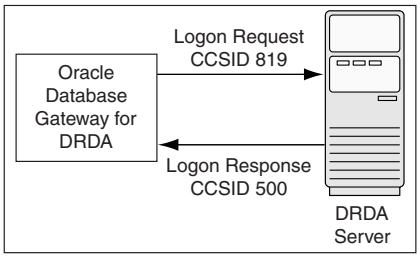
During logon of the gateway to the DRDA Server, initial language information is exchanged between the Gateway and the server. First, the gateway sends to the DRDA Server the CCSID it will be conversing in. In the following example, the Oracle character set "WE8ISO8859P1" is mapped to CCSID 819 (an ASCII Code Page). This CCSID is sent to the DRDA Server. The DRDA Server responds with the CCSID that it will be conversing in. This will be the CCSID with which the DB2 database was generated. Also, in the following example, this is CCSID 500, an EBCDIC code page. Figure E–2, "Gateway Language Interaction with DRDA Server", illustrates this process.

A DB2 instance will map unknown CCSIDs using the SYSIBM.SYSSTRINGS table (this table has different names for the various DB2 versions). It is possible to add additional character set mappings to this table using DB2 utilities. Please refer to the DB2 Installation documentation for details.

The setting of the HS_LANGUAGE parameter in the gateway init*sid*.ora determines which CCSID is used by the gateway for the conversation. Similarly, the setting of the HS_NLS_NCHAR parameter determines which CCSID will be used by the gateway for GRAPHIC data interchange. For the list of supported ASCII-based Oracle character sets that are mapped to CCSIDs, refer to "Gateway Codepage Map Facility" on page E-5.

Note again that the gateway character set need not be the same as the Oracle database character set. In many cases, it is not feasible to set the gateway character set equal to the Oracle database character set because the DRDA Server will not have a valid translation for it. Instead, choose a character set that will have the most complete intersection with the character set that is used by the DRDA Server. The Oracle Net facility will do any translation between the gateway character set and the Oracle database character set.

Figure E–2 Gateway Language Interaction with DRDA Server



Gateway Configuration

After the gateway is installed, you must change several parameters to customize for Globalization Support support.

Globalization Support Parameters in the Gateway Initialization File

Four parameters in the gateway initialization file (initsid.ora) affect Globalization Support:

- HS_LANGUAGE
- HS_NLS_NCHAR
- HS_NLS_DATE_FORMAT
- HS_NLS_DATE_LANGUAGE

HS_LANGUAGE

HS_LANGUAGE defines the character set that is used for communication between the gateway and the DRDA Server. It specifies the conventions such as: the language used for messages from the target system; names of days and months; symbols for AD, BC, AM, and PM; and default language sorting mechanism.

The syntax of the HS_LANGUAGE parameter is:

```
HS_LANGUAGE=language[_territory.character_set]
```

where:

language can be any valid language.

territory is optional, and defaults to AMERICA.

character_set is optional and defaults to US7ASCII. This must be an ASCII base character set name, and it should match a character set listed in the gateway codepage map. Refer to "Gateway Codepage Map Facility" on page E-5 for the list of supplied character set mappings.

If you omit the HS_LANGUAGE parameter from **init*sid*.ora**, then the default setting is AMERICAN_AMERICA.US7ASCII. EBCDIC character sets are not supported. The values for *language* and *territory* (such as AMERICAN_AMERICA) must be valid, but they have no effect on translations.

HS_NLS_NCHAR

See "HS_NLS_NCHAR" on page C-25 for more information on the parameter.

HS_NLS_DATE_FORMAT

HS_NLS_DATE_FORMAT specifies the format for dates used by the DRDA Server.

The syntax of the NLS_DATE_FORMAT parameter is:

```
HS_NLS_DATE_FORMAT=date_format
```

where *date_format* must be YYYY-MM-DD, the ISO date format. If this parameter is set to any other value or is omitted, then you receive an error when updating, deleting from, selecting from, or inserting into, a table with date columns.

HS_NLS_DATE_LANGUAGE

HS_NLS_DATE_LANGUAGE specifies the language used by the DRDA Server for day and month names, and for date abbreviations. Because ISO date format contains numbers only, this parameter has no effect on gateway date processing and should be omitted.

Gateway Codepage Map Facility

The gateway now has a user-specifiable facility to map IBM Coded Character Set Identifiers (CCSIDs) to Oracle Character Sets for data translation.

The map name defaults to codepage.map and is located in the directory *ORACLE_HOME*\tg4drda\admin. Refer to Appendix C, "Initialization Parameters" for more detailed information about the DRDA_CODEPAGE_MAP parameter.

The map has two different forms of syntax. The first form of syntax defines a mapping between a CCSID and an Oracle database character set:

```
[S|D|M] CCSID direction Oracle_CharacterSet {shift}
```

where:

`S` designates a single-byte character set

`D` designates a double-byte character set

`M` designates a multibyte character set

`CCSID` is the IBM coded character set identifier

`direction` is one of the following:

- `=` means mapping is bidirectional
- `<` means mapping is one-way, Oracle character set to CCSID
- `>` means mapping is one-way, CCSID to Oracle character set

`Oracle_CharacterSet` is the name of a valid Oracle character set.

`shift` indicates a character set that requires `Shift OUT/IN` processing. Set this attribute only for EBCDIC-based double-byte and multibyte mappings.

The second form of syntax defines a mapping of a multibyte CCSID to its single-byte and double-byte CCSID equivalents:

```
MBC multi = single double
```

where:

`multi` is the multibyte CCSID

`single` is the single-byte CCSID

`double` is the double-byte CCSID

This facility is intended as a way of mapping CCSIDs which were not previously mapped as shipped with the gateway. You must contact Oracle Support Services before modifying this map.

The following are the contents of the map as shipped with the Oracle Database Gateway for DRDA;

```
# Copyright (c) 2001, 2003, Oracle Corporation.  All rights reserved.
# Transparent Gateway for IBM DRDA - CodePage/Oracle CharacterSet Map
# S==Single-byte, D==Double-byte, M==Multi-byte, MBC==SBC DBC mapping
#
# Single-byte codepage mappings
#
S    37 = WE8EBCDIC37   # United States/Canada                          EBCDIC
S   273 = D8EBCDIC273   # Austria/Germany                               EBCDIC
S   277 = DK8EBCDIC277  # Denmark/Norway                                EBCDIC
S   278 = S8EBCDIC278   # Finland/Sweden                                EBCDIC
S   280 = I8EBCDIC280   # Italy                                         EBCDIC
S   284 = WE8EBCDIC284  # Latin America/Spain                           EBCDIC
S   285 = WE8EBCDIC285  # United Kingdom                                EBCDIC
S   297 = F8EBCDIC297   # France                                        EBCDIC
#S   420 = AR8EBCDICX    # Arabic Bilingual (USA English)               EBCDIC
S   420 = AR8XBASIC     # Arabic Bilingual (USA English)               EBCDIC
S   424 = IW8EBCDIC424  # Israel (Hebrew)                              EBCDIC
S   437 = US8PC437      # Personal Computer,USA                        ASCII
S   500 = WE8EBCDIC500  # International                                 EBCDIC
S   813 = EL8ISO8859P7  # Greek                                        ASCII
S   819 = WE8ISO8859P1  # ISO/ANSI Multilingual                        ASCII
S   838 = TH8TISEBCDIC  # Thai w/Low-Tone Marks & Ancient Chars        EBCDIC
```

```
S   850 < US7ASCII      # Multilingual Page - Personal Computer         ASCII
S   850 = WE8PC850      # Multilingual Page - Personal Computer         ASCII
S   864 = AR8ISO8859P6  # Arabic - Personal Computer                    ASCII
S   870 = EE8EBCDIC870  # Latin 2, Multilingual/ROECE                   EBCDIC
S   871 = WE8EBCDIC871  # Iceland - CECP                                EBCDIC
S   875 = EL8EBCDIC875  # Greece                                        EBCDIC
S   904 > US7ASCII      # Traditional Chinese - PC-Data                 ASCII
S   912 = EE8ISO8859P2  # Latin 2 8-bit                                 ASCII
S   916 = IW8ISO8859P8  # Israel (Hebrew)                               ASCII
S  1025 = CL8EBCDIC1025 # Cyrillic, Multiling                           EBCDIC
S  1086 = IW8EBCDIC1086 # Israel                                        EBCDIC
S  1252 = WE8MSWIN1252  # Latin 1 - MS-Windows                          ASCII
S  1253 = EL8MSWIN1253  # Greek - MS-Windows                            ASCII
S 28709 > WE8EBCDIC37   # United States/Canada (CP28709==CP37)          EBCDIC
#
# Multibyte codepage mappings
#
#S   833 > KO16DBCS       # Korean Extended single-byte                 EBCDIC
#D   834 > KO16DBCS shift # Korean double-byte                          EBCDIC
#M   933 = KO16DBCS shift # Korean Mixed multi-byte                     EBCDIC

#MBC 933 = 833 834        # Korean Mixed multi-byte                     EBCDIC
#
#S  1088 > KO16MSWIN949 # Korean KS single-byte PC-Data                 ASCII
#D   951 > KO16MSWIN949 # Korean KS double-byte PC-Data                 ASCII
#M   949 = KO16MSWIN949 # Korean KS multi-byte PC-Data                  ASCII
#MBC 949 = 1088 951     # Korean KS multi-byte PC-Data                  ASCII
#
#S   891 > KO16KSC5601 # Korean single-byte                            ASCII
#S  1040 > KO16KSC5601 # Korean single-byte                            ASCII
#D   926 > KO16KSC5601 # Korean double-byte                            ASCII
#M   934 = KO16KSC5601 # Korean multi-byte                             ASCII
#M   944 > KO16KSC5601 # Korean multi-byte                             ASCII
#MBC 934 = 891 926     # Korean multi-byte                             ASCII
#MBC 944 = 1040 926    # Korean multi-byte Extended                    ASCII
#
#S 28709 > ZHT16DBCS       # Traditional Chinese single-byte           EBCDIC
#D   835 > ZHT16DBCS shift # Traditional Chinese double-byte           EBCDIC
#M   937 = ZHT16DBCS shift # Traditional Chinese multi-byte            EBCDIC
#MBC 937 = 28709 835       # Traditional Chinese multi-byte            EBCDIC
#
#S  1114 > ZHT16MSWIN950 # Traditional Chinese single-byte             ASCII
#D   947 > ZHT16MSWIN950 # Traditional Chinese double-byte             ASCII
#M   950 = ZHT16MSWIN950 # Traditional Chinese multi-byte              ASCII
#MBC 950 = 1114 947      # Traditional Chinese multi-byte              ASCII
#

#S   836 > ZHS16DBCS       # Simplified Chinese single-byte            EBCDIC
#D   837 > ZHS16DBCS shift # Simplified Chinese double-byte            EBCDIC
#M   935 = ZHS16DBCS shift # Simplified Chinese multi-byte             EBCDIC
#MBC 935 = 836 837         # Simplified Chinese multi-byte             EBCDIC
#
#S  1027 > JA16DBCS       # Japanese single-byte                       EBCDIC
#D   300 > JA16DBCS shift # Japanese double-byte                       EBCDIC
#D  4396 > JA16DBCS shift # Japanese double-byte                       EBCDIC
#M   939 = JA16DBCS shift # Japanese multi-byte                        EBCDIC
#M  5035 > JA16DBCS shift # Japanese multi-byte                        EBCDIC
#MBC  939 = 1027 300      # Japanese multi-byte                        EBCDIC
#MBC 5035 = 1027 4396     # Japanese multi-byte                        EBCDIC
#
```

```
#S   290 > JA16EBCDIC930       # Japanese single-byte                  EBCDIC
#D   300 > JA16EBCDIC930 shift # Japanese double-byte                  EBCDIC
#D  4396 > JA16EBCDIC930 shift # Japanese double-byte                  EBCDIC
#M   930 = JA16EBCDIC930 shift # Japanese multi-byte                   EBCDIC
#M  5026 > JA16EBCDIC930 shift # Japanese multi-byte                   EBCDIC
#MBC  930 = 290 300            # Japanese multi-byte                   EBCDIC
#MBC 5026 = 290 4396           # Japanese multi-byte                   EBCDIC
#
```

Refer to the following list to check the character set of an existing database:

- **for DB2/OS390:** Ask your system administrator. There is no single command you use.
- **for DB2/400:** Run the command `DSPSYSVAL SYSVAL(QCCSID)`
- **for DB2/UDB:** Ask your system administrator. There is no single command you use.

Multibyte and Double-Byte Support in the Gateway

To enable the gateway to properly handle double-byte and multibyte data, you must configure the codepage map facility with proper multibyte maps and (as an option) you can set the following gateway configuration parameters:

- DRDA_GRAPHIC_LIT_CHECK
- DRDA_GRAPHIC_TO_MBCS
- DRDA_MBCS_TO_GRAPHIC
- DRDA_GRAPHIC_PAD_SIZE
- DRDA_GRAPHIC_CHAR_SIZE

Refer to Appendix C, "Initialization Parameters", for the values of these parameters.

Configuring the codepage map requires knowledge of the codepages that have been configured in the DRDA Server database as well as knowledge of compatible Oracle database character sets.

IBM coded character set identifiers (CCSIDs) are used to indicate which codepages are configured as the primary codepage for the database, as well as any translation character sets loaded into the database. Some DRDA Servers, such as with DB2, have a translation facility in which character set transforms are mapped between two compatible character sets. For DB2/OS390, these transforms are stored in the table SYSIBM.SYSSTRINGS and transform on the CCSID codepage to another CCSID codepage. In SYSSTRINGS, IN and OUT columns specify the CCSIDs that are used in the transform. Typical transforms are from ASCII to EBCDIC and back again. Two transforms are therefore used for two given CCSIDs.

Multibyte codepages are a composite of a single-byte codepage and a double-byte codepage. As an example, the Korean EBCDIC multi-byte codepage, CCSID 933, is composed of two codepages, codepage 833 (for single-byte) and codepage 834 (for double-byte). The DRDA Server, therefore, can send data to the gateway in any of these three codepages, and the gateway must translate suitably depending on which codepage the data is associated with. Because CCSID 933 is an EBCDIC-based codepage, and the gateway must use an ASCII-based codepage, we identify an equivalent set of codepages, which are ASCII-based. An example would be the Korean multibyte codepage, CCSID 949, which is composed of two codepages, codepage 1088 (for single-byte) and codepage 951 (for double-byte).

The codepage map facility is used to map these CCSIDs into the equivalent Oracle database character sets. Unlike IBM CCSIDs, Oracle database character sets are unified (in that single-byte and double-byte character sets have been combined into one set) and are thus identified by one ID instead of three IDs. In our previous example, the equivalent Oracle database character set for the ASCII Korean codepages would be KO16MSWIN949, and the EBCDIC Korean codepages would be KO16DBCS. These are identified to the gateway by using a set of mapping entries in the **codepage.map** file.

First, the EBCDIC Korean sets are:

```
S   833 > KO16DBCS       # Korean Extended single-byte                EBCDIC
D   834 > KO16DBCS shift # Korean double-byte                         EBCDIC
M   933 = KO16DBCS shift # Korean Mixed multi-byte                    EBCDIC
MBC 933 = 833 834        # Korean Mixed multi-byte                    EBCDIC
```

Notice that the multibyte set is a bidirectional map to KO16DBCS, while the single and double codepages are mapped one-way to KO16DBCS. Because only one bidirectional CCSID to Oracle database character set entry for a given pair can exist, we directly map the multibyte sets. And because the single-byte and double-byte CCSIDs are ostensibly subsets of KO16DBCS, we map them as one-way entries. Note that double-byte and multibyte maps are tagged with the shift attribute. This is required for EBCDIC double-byte and multibyte codepages as part of the shift out/in encapsulation of data. Note that the single-byte map is not marked because single-byte sets are not permitted to contain double-byte data and thus will never use shift encapsulation. Also note that the MBC entry ties the codepages together.

The ASCII Korean sets are similarly mapped and are:

```
S  1088 > KO16MSWIN949 # Korean KS single-byte PC-Data               ASCII
D   951 > KO16MSWIN949 # Korean KS double-byte PC-Data               ASCII
M   949 = KO16MSWIN949 # Korean KS multi-byte PC-Data                ASCII
MBC 949 = 1088 951     # Korean KS multi-byte PC-Data                ASCII
```

Notice that the multibyte set is a bidirectional map to KO16MSWIN949, while the single and double codepages are mapped one-way to KO16MSWIN949. Because only one bidirectional CCSID to Oracle database character set entry for a given pair can exist, we directly map the multibyte sets. And because the single-byte and double-byte CCSIDs are ostensibly subsets of KO16MSWIN949, we map them as one-way entries. Note that there is no shift attribute in any of these mappings. This is because ASCII-based sets do not use shift out/in encapsulation. Instead, ASCII-based sets use a different method (which does not use a shift out/in protocol) to identify double-byte characters.

The preceding entries supply the necessary codepage mappings for the gateway. To complete the example, we need to specify the correct character set in the HS_ LANGUAGE and HS_NLS_NCHAR parameters in the gateway initialization file. The gateway initialization parameters would look as follows:

```
HS_LANGUAGE=AMERICAN_AMERICA.KO16MSWIN949
HS_NLS_NCHAR=KO16MSWIN949
```

Note that the specified character set must be ASCII-based.

This takes care of configuration of the gateway. The last step is to set up transforms between the EBCDIC codepages and the ASCII codepages in the DRDA Server database. Normally, the gateway would use a total of six transforms, one of each pair in both directions. You may save some table space by installing only the ASCII-to-EBCDIC transforms. The reasoning is that the DRDA Server needs to translate only the ASCII data that is sent by the gateway, but the DRDA Server does not need to send ASCII data. The gateway will receive the EBCDIC data and translate

as needed. This one-sided data transfer methodology is called "receiver-makes-right", meaning that the receiver must translate whatever character set the sender uses. In our example, the DRDA Server is EBCDIC-based, so it will send all data in EBCDIC. The server, therefore, does not need to have an EBCDIC-to-ASCII transform because the server will never use the transform.

In our previous example, the DRDA Server database is assumed to be EBCDIC, which is likely to be true for a DB2/OS390 database. For a DB2/UDB database, however, this is not likely to be true. Because most DB2/UDB databases are running on ASCII-based computers, they will likely be created with ASCII-based codepages. In such cases, the gateway needs to have only one set of codepage map definitions, which are those for the ASCII set. Also, because both the DRDA Server and the gateway will be using the same codepages, no character set transforms need to be loaded into the DB2 database. This can help reduce the amount of CPU overhead that is associated with character translation.

One final note concerning codepage map entries: Be aware that some multi-byte codepages may be composed of single-byte CCSIDs that are already defined in the codepage.map file that is provided with the product. If you are adding a new set of entries to support a multibyte set, then comment out the provided entries so that your new entries will be used correctly.

Additional codepage mappings, which are not already provided, are possible. You may construct entries such as those in our examples, given knowledge of the IBM CCSIDs and the Oracle database character sets. Because this can be complex (given the IBM documentation of codepage definitions and Oracle database character set definitions), thoroughly test your definitions for all desired character data values before putting them into production.

If you are uncertain, then contact Oracle Support Services to request proper codepage mapping entries.

Message Availability

Whether a language message module is available depends on which modules are installed in the Oracle product set running on the server. If message modules for a particular language set are not installed, then specifying that language with a language parameter does not display messages in the requested language.

Example of Globalization Support Configuration

Following is an example of all the settings needed to configure the gateway, DRDA Server, Oracle database, and client so that a language and character set are working compatibly across the system. In this example, the settings enable a customer in Germany to interact with the gateway in German:

Gateway init*sid*.ora file:

```
HS_LANGUAGE=AMERICAN_AMERICA.WE8ISO8859P1
HS_NLS_DATE_FORMAT=YYYY-MM-DD
```

DRDA Server CCSID:

```
273 (D8EBCDIC273)
```

Oracle Database and client setting for database:

```
SELECT USERENV('language') FROM DUAL;
USERENV('LANGUAGE')
```

```
----------------------------
AMERICAN_AMERICA.WE8ISO8859P1
```

Oracle Database and client environment variables:

```
NLS_LANG=GERMAN_GERMANY.WE8ISO8859P1
```

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U