Oracle® Transparent Gateway for Informix

Administrator's Guide

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

B14274-01

June 2005

ORACLE®

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

B14274-01

Copyright © 2003, 2005, Oracle. All rights reserved.

Primary Author: Amitai Sela

Contributing Author: Laurel Hale, Cynthia Kibbe, Kishan Peyetti

Contributor: Orit Curiel, Jacco Draaijer, Vira Goorah

Contents

3   Informix Gateway Features and Restrictions

4 Case Studies

A Data Type Conversion

B Supported SQL Syntax and Functions

C Data Dictionary

D   Heterogeneous Services Initialization Parameters

Index

List of Tables

Send Us Your Comments

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

B14274-01

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

- Did you find any errors?

- Is the information clearly presented?

- Do you need more information? If so, where?

- Are the examples correct? Do you need more examples?

- What features did you like most about this manual?

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

- Electronic mail: infodev_us@oracle.com

- FAX: (650) 506-7227.   Attn: Server Technologies Documentation Manager

- Postal service:

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

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

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

x

Preface

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

This preface covers the following topics:

- Audience
- Documentation Accessibility
- Organization
- Related Documentation
- Conventions

Audience

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

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

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

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

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

Organization

This document contains:

Chapter 1, "Introduction"

This chapter introduces the Oracle Transparent Gateway for Informix.

Chapter 2, "Configuring the Gateway"

This chapter explains how to configure the gateway for Informix.

Chapter 3, "Informix Gateway Features and Restrictions"

This chapter explains how to use the gateway to access Informix data, pass Informix commands from applications to the Informix database, perform distributed queries, and copy data.

Chapter 4, "Case Studies"

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

Appendix A, "Data Type Conversion"

This appendix describes how the gateway converts Informix data types to Oracle data types.

Appendix B, "Supported SQL Syntax and Functions"

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

Appendix C, "Data Dictionary"

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

Appendix D, "Heterogeneous Services Initialization Parameters"

This appendix contains information about Heterogeneous Services initialization parameters.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Database New Features*

- *Oracle Call Interface Programmer's Guide*

- *Oracle Enterprise Manager Administrator's Guide*

- *Oracle Database Administrator's Guide*

- *Oracle Database Application Developer's Guide - Fundamentals*

- *Oracle Database Concepts*

- *Oracle Database Performance Tuning Guide*

- *Oracle Database Error Messages*

- *Oracle Database Globalization Support Guide*

- *Oracle Database Reference*

- *Oracle Database SQL Reference*

- *Oracle Database Net Services Administrator's Guide*

- *SQL*Plus User's Guide and Reference*

- *Oracle Database Heterogeneous Connectivity Administrator's Guide*

- *Oracle Database 2 Day DBA*

- *Oracle Database Security Guide*

Many of the examples in this book use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

http://oraclestore.oracle.com/

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

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

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

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

Conventions

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

- Conventions in Text

-

Conventions in Text

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

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

Conventions in Code Examples

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

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

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

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

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

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

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

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

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

1

Introduction

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

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

This chapter contains the following sections:

- Overview
- Heterogeneous Services Technology
- Oracle Transparent Gateways

Overview

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

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

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

Heterogeneous Services Technology

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

Heterogeneous Services extend the Oracle SQL engine to recognize the SQL and procedural capabilities of the remote non-Oracle system and the mappings required to obtain necessary data dictionary information. Heterogeneous Services provides two types of translations: the ability to translate Oracle SQL into the proper dialect of the non-Oracle system as well as data dictionary translations which displays the metadata of the non-Oracle system in the local format. For situations where no translations are available, native SQL can be issued to the non-Oracle system using the pass-through feature of Heterogeneous Services.

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

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

Oracle Transparent Gateways

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

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

2

Configuring the Gateway

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

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

Configuring the Gateway

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

Task 1: Choose a System Identifier for the Gateway

The gateway system identifier (SID) is an alphanumeric character string that identifies a gateway instance. You need one gateway instance, and therefore one gateway SID, for each 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 one Informix 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 Informix database.

Task 2: Customize the Initialization Parameter File

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

```
$ORACLE_HOME/tg4ifmx/admin/inittg4ifmx.ora
```

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

If you are not using tg4ifmx as the gateway SID, you must rename the initialization parameter file using the SID you chose in Task 1. This default initialization parameter

file is sufficient for starting the gateway, verifying a successful installation, and running the demonstration scripts.

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

```
HS_FDS_CONNECT_INFO=database_name@server_name
```

If you specify only *database_name*, omitting *server_name*, the gateway binds to the database residing on the local Informix server (as specified in the environment variable INFORMIXSERVER).

Additionally, set the Informix environment variable, as follows:

```
SET INFORMIXDIR=ifmx_clt_dir
```

> **Note:** The *ifmx_clt_dir* specified is the directory where the client library is installed.

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

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

Configuring Oracle Net Services Listener for the Gateway

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

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

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

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

> **Note:** If Oracle Net Services is reinstalled, the original listener.ora file is renamed and a new listener.ora file is put into the $ORACLE_HOME/network/admin directory.

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

- A list of Oracle Net Services addresses on which the TNS listener listens

- The gateway that the TNS listener starts in response to incoming connection requests

Example of Address to Listen On in listener.ora File

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

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

Where:

Table 2–1   Parameters for listener.ora File

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

Example of Gateway to Start in listener.ora File

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

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
          (SID_NAME=gateway_sid)
          (ORACLE_HOME=oracle_home_directory)
          (PROGRAM=tg4ifmx)
          (ENVS=SHLIB_PATH=ifmx_clt_dir/lib:ifmx_clt_dir/lib/esql:        oracle_
home_directory/lib32)
      )
   )
```

Where:

Table 2–2   Parameters for listener.ora File

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 Oracle Transparent Gateway for Informix.
ifmx_clt_dir	specifies the value of the INFORMIXDIR environment variable.

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

```
SID_LIST_LISTENER=
(SID_LIST=
   (SID_DESC=.
```

```
      .
    )
    (SID_DESC=.
      .
    )
    (SID_DESC=
       (SID_NAME=gateway_sid)
       (ORACLE_HOME=oracle_home_directory)
       (PROGRAM=tg4ifmx)
       (ENVS=SHLIB_PATH=ifmx_clt_dir/lib:ifmx_clt_dir/lib/esql:        oracle_home_
directory/lib32)
    )
)
```

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

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

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

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

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

    If you have the C Shell, enter the following:

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

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

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

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

    ```
    $ lsnrctl status
    ```

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

```
LSNRCTL for HPUX: Version 10.2.0.1.0 - Production on 01-JUN-2005 09:16:17

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

Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=204.179.99.15)(PORT=1521))
STATUS of the LISTENER
---------------------
Alias                     listener
Version                   TNSLSNR for HPUX: Version 10.2.0.1.0 - Production
Start Date                31-May-2005 10:16:17
Uptime                    0 days 23 hr. 0 min. 0 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File   /users/oracle/gateway/network/admin/listener.ora
Listener Log File         /users/oracle/gateway/network/log/listener.log
Listening Endpoints Summary...
```

```
    (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=204.179.99.15)(PORT=1521)))
Services Summary...
Service "tg4ifmx" has 1 instance(s).
  Instance "tg4ifmx", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

In this example, `tg4ifmx` is the default SID value assigned during installation. You can use any valid ID for the SID, or keep the default.

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

Configuring the Oracle Database Server for Gateway Access

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

Configuring Oracle Net Services for the Oracle Database Server

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

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

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

Configuring tnsnames.ora

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

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

Where:

Table 2–3   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 Services TNS listener that is listening for the gateway. The TNS listener's port number can be found in the `listener.ora` file used by the TNS listener. See "Example of Address to Listen On in listener.ora File" on page 2-3.
gateway_sid	specifies the SID of the gateway and matches the SID specified in the `listener.ora` file of the TNS listener that is listening for the gateway. See "Task 1: Configure Oracle Net Services TNS Listener for the Gateway" on page 2-2 for more information.
`(HS=OK)`	specifies that this connect descriptor uses the Oracle Heterogeneous Services option.

Creating Database Links

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

A connection to the gateway is established through a database link when it is first used in an Oracle session. In this context, a connection refers to the connection between the Oracle database server and the gateway. The connection remains established until the Oracle session ends. Another session or user can access the same database link and get a distinct connection to the gateway and 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. The database and application administrators of a distributed database system are responsible for managing the necessary database links that define paths to the Informix database.

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

Gateway Password Encryption Tool

The gateway uses userids and passwords to access the information in the remote database. Some userids and passwords must be defined in the Gateway Initialization File to handle functions such as resource recovery. In the current security conscious environment, having plain-text passwords that are accessible in the Initialization File is deemed insecure. The `tg4pwd` encryption utility has been added as part of Heterogeneous Services' generic connectivity to help make this more secure. This

utility is accessible by this gateway. The initialization parameters which contain sensitive values can be stored in an encrypted form.

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

Configuring the Gateway for Multiple 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 server named inf74_hpux are being added.

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

- Configuring the gateway.
- Configuring the Informix environment.
- Configuring Oracle Net Services for the gateway and the Oracle database server.

Multiple Databases Example: Configuring the Gateway

Choose Two System IDs for Each Informix Database

A separate instance of the gateway accesses the different Informix databases. 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
$ cp inittg4ifmx.ora inittg4ifmx2.ora
$ cp 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=db2@inf74_hpux
```

For inittg4ifmx3.ora, enter the following:

```
HS_FDS_CONNECT_INFO=db3@inf74_hpux
```

> **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 Databases Example: Configuring the Informix Environment

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

For `inittg4ifmx2.ora`, enter the following:

```
SET INFORMIXDIR=ifmx_clt_directory
```

For `inittg4ifmx3.ora`, enter the following:

```
SET INFORMIXDIR=ifmx_clt_directory
```

The *ifmx_clt_directory* specified is the directory where the client library is installed.

Multiple Databases Example: Configuring Oracle Net Services Listener

Add Entries to listener.ora

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

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

```
SID_LIST_LISTENER=
(SID_LIST=
   (SID_DESC=
      (SID_NAME=tg4ifmx)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4ifmx)
   (ENVS=SHLIB_PATH=ifmx_clt_dir/lib:ifmx_clt_dir/lib/esql:
oracle_home_directory/lib32)
   )
   (SID_DESC=
      (SID_NAME=tg4ifmx2)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4ifmx)
      (ENVS=SHLIB_PATH=ifmx_clt_dir/lib:ifmx_clt_dir/lib/esql:oracle_home_
directory/lib32)
   )
   (SID_DESC=
      (SID_NAME=tg4ifmx3)
      (ORACLE_HOME=oracle_home_directory)
      (PROGRAM=tg4ifmx)
   (ENVS=SHLIB_PATH=ifmx_clt_dir/lib:ifmx_clt_dir/lib/esql:
oracle_home_directory/lib32)
   )
)
```

Multiple Databases Example: Stopping and Starting the TNS Listener

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

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

Multiple Databases Example: Configuring for Gateway Access

Configuring Oracle Net Services for Multiple Gateway Instances

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

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

```
old_db_using=(DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                    (SID=tg4ifmx))
              (HS=OK))
new_db2_using=(DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                    (SID=tg4ifmx2))
              (HS=OK))
new_db3_using=(DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (PORT=1541)
                (HOST=gtwhost))
                (CONNECT_DATA=
                    (SID=tg4ifmx3))
              (HS=OK))
```

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

Multiple Databases Example: Accessing 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';
```

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

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

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

Or

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

Performing Configuration Tasks

You can perform the following configuration tasks:

- Configuring for Two-Phase Commit

Configuring for Two-Phase Commit

The gateway supports the following transaction capabilities:

- COMMIT_CONFIRM
- READ_ONLY
- SINGLE_SITE

By default, the gateway runs in COMMIT_CONFIRM transaction mode. When the
Informix database is updated by a transaction, the gateway becomes the commit point
site. The Oracle database server 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 can participate in an Oracle two-phase commit
transaction as the commit point site.

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

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

Task 1: Create a Recovery Account and Password

For the gateway to recover distributed transactions, a recovery account and password
must be set up in the 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 Corporation recommends that you do **not** use the default value RECOVER for the user name and password. Moreover, storing plian 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:** "Task 2: Customize the Initialization Parameter File" on page 2-1 for information about editing the initialization parameter file. For information about HS_FDS_RECOVERY_ACCOUNT and HS_FDS_RECOVERY_PWD, see Appendix D, "Heterogeneous Services Initialization Parameters".

Task 2: Create the Transaction Log Table

When configuring the gateway for two-phase commit, a table must be created in the 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 D, "Heterogeneous Services Initialization Parameters" for information about the HS_FDS_TRANSACTION_ LOG initialization parameter.

Create the transaction log table in the user account you created in "Task 1: Create a Recovery Account and Password" on page 2-10. Because the transaction log table is used to record the status of a gateway transaction, the table must reside at the database where the 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, as follows:

1. Login as userid RECOVER.

2. Set environment variable DELIMIDENT.

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

   ```
   $ DELIMIDENT = y; export DELIMIDENT
   ```

   If you have the C Shell, enter the following:

   ```
   $ setenv DELIMIDENT y
   ```

3. Execute the script using dbaccess, as follows.

   ```
   $ cd $ORACLE_HOME/tg4ifmx/admin
   $ dbaccess database_name tg4ifmx_tx.sql
   ```

3

Informix Gateway Features and Restrictions

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

This chapter contains the following sections:

- Using the Pass-Through Feature
- Database Compatibility Issues for Informix
- Known Restrictions
- Known Problems

Using the Pass-Through Feature

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

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

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

Where *command* cannot be one of the following:

- `COMMIT`
- `CREATE DATABASE`
- `DROP DATABASE`
- `ROLLBACK`
- `ROLLFORWARD DATABASE`
- Informix tool commands

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

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

Database Compatibility Issues for Informix

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

- ANSI SQL Standard
- Naming Rules
- Data Types
- Queries
- Locking

ANSI SQL Standard

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. The gateway supports only Informix databases that comply with the ANSI standard. For more information about how to create or start up an ANSI-compliant Informix database, refer to your Informix documentation.

Naming Rules

Naming rule issues include the following:

- Rules for Naming Objects
- Object Names
- Case Sensitivity

Rules for Naming Objects

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

See Also: *Oracle Database Reference* and Informix documentation.

Object Names

Names of Informix database objects are limited to a maximum of 18 characters. An object name can be composed of these characters:

- Numbers 0 to 9
- Lowercase letters a to z
- Uppercase letters A to Z
- Underscore character (_)

Case Sensitivity

Informix handles letter case differently from Oracle. Informix uses these rules:

- Table owner names default to uppercase letters, unless the name is surrounded by double quote characters

- Column names, table names, view names, and so on, are always treated as lowercase letters

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

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

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

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

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

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

Or

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

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

If existing applications cannot be changed according to these conventions, create views in Oracle to associate Informix names to the correct letter case. For example, to refer to the Informix table emp from an existing Oracle application by using only uppercase names, define the following view:

```
SQL> CREATE VIEW EMP (EMPNO, ENAME, SAL, HIREDATE)
     AS SELECT "empno", "ename", "sal", "hiredate"
     FROM "emp"@IFMX;
```
With this view, the application can issue statements such as the following:

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

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

> **Note:** The DELIMIDENT Informix parameter must be set in the Informix client in order for Informix to accept double quotes around object names.

Data Types

Data type issues include the following:

- Binary, Byte and Text Literal Notation
- Data Type Conversion

Binary, Byte and Text Literal Notation

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

This notation is not converted to syntax compatible with Informix BINARY, BYTE and TEXT data types (a 0x followed by hexadecimal digits, surrounded by single quotes).

For example, the following statement is not supported:

```
SQL> INSERT INTO BYTE_TAB@IFMX VALUES ('Oxff');
```

Where BYTE_TAB contains a column of data type BINARY, BYTE or TEXT. Use bind variables when inserting into or updating BINARY, BYTE or TEXT data types.

Data Type Conversion

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

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

```
SELECT DATE_COL FROM TEST@
IFMX
 WHERE DATE_COL = "1-JAN-2001";
```

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

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

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

Queries

Query issues include the following:

- Row Selection
- Empty Strings
- Empty Bind Variables

Row Selection

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

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

Empty Strings

Oracle processes an empty string in a SQL statement as a null value. Informix processes an empty string as an empty string.

Comparing to an empty string

The Gateway passes literal empty strings to the Informix database without any conversion. If you intended an empty string to represent a null value, Informix does not process the statement that way; it uses the empty string.

You can avoid this problem by using NULL or IS NULL in the SQL statement instead of the empty string syntax, as in the following example:

```
SELECT * from "emp"@IFMX where "ename" IS NULL;
```

Selecting an empty string

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

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

Empty Bind Variables

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

Locking

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

- Read access might block write access

- Write access might block read access

- Statement-level read consistency is not guaranteed

> **See Also:** Informix documentation for information about the Informix locking model.

Known Restrictions

If you encounter incompatibility problems not listed in this section or in "Known Problems" on page 3-8, please contact Oracle Support Services. The following section describes the known restrictions and includes suggestions for dealing with them when possible:

- Transactional Integrity

- Transaction Capability

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

- Pass-Through Feature

- Informix SMALLFLOAT and REAL Data Types

- Informix NCHAR and NVARCHAR Data Types

- SQL Syntax

- SQL*Plus

- Database Links

- Gateway Data Dictionary Views

- Stored Procedures

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

Transactional Integrity

The gateway cannot guarantee transactional integrity in the following cases:

- When a statement that is processed by the gateway causes an implicit commit in the target database

- When the target database is configured to work in autocommit mode

> **Note:** Oracle corporation strongly recommends the following:
>
> - If you know that executing a particular statement causes an implicit commit in the target database, then ensure that this statement is executed in its own transaction.
>
> - Do not configure the target database to work in autocommit mode.

Transaction Capability

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

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

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

Informix version 7.23 has a bug which prevents configuring the gateway as TWO_PHASE_COMMIT.

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

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

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

```
ORA-1002:  fetch out of sequence
```

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

Pass-Through Feature

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

Informix SMALLFLOAT and REAL Data Types

Informix SMALLFLOAT and REAL data types have a precision of 6.

Informix NCHAR and NVARCHAR Data Types

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

SQL Syntax

This section lists restrictions on the following SQL syntax:

- WHERE CURRENT OF Clause
- CONNECT BY Clause
- Use of NULL Keyword in SELECT Statement
- Subqueries in INSERT Statement
- Subqueries in DELETE, INSERT, and UPDATE Statements
- ROWID
- EXPLAIN PLAN Statement

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

WHERE CURRENT OF Clause

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

CONNECT BY Clause

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

Use of NULL Keyword in SELECT Statement

The NULL keyword cannot be used in the select list of a SELECT statement because that syntax is not ANSI SQL.

For example, the following statement cannot be used:

```
SQL> SELECT NULL FROM ...
```

Subqueries in INSERT Statement

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

```
SQL> INSERT INTO "emp_target"@IFMX
        SELECT a."empno" FROM "emp_source"@IFMX a,
```

```
"emp_source"@IFMX b WHERE b."empno"=9999
```

Subqueries in DELETE, INSERT, and UPDATE Statements

SQL statements in subqueries of DELETE, INSERT, and UPDATE statements cannot refer to the same table as specified in the outer query. This is because of the locking mechanism in Informix.

ROWID

The Oracle ROWID implementation is not supported.

EXPLAIN PLAN Statement

The EXPLAIN PLAN statement is not supported.

SQL*Plus

In SQL*Plus, the gateway does not support using a SELECT statement to retrieve data from an Informix column defined as data type BYTE.

You need to use double quotes to wrap around lowercase table names, for example:

```
copy from tkhouser/tkhouser@inst1 insert loc_tkhodept using select* from
"tkhodept"@holink2;
```

Database Links

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

Gateway Data Dictionary Views

Only the first 64 characters of the view definition are returned when querying ALL_VIEWS and USER_VIEWS in the gateway data dictionary.

Stored Procedures

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

Known Problems

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

The following known problems are described in this section:

- Encrypted Format Login

- Informix BYTE and TEXT Data Types

- Schema Names and PL/SQL

- Data Dictionary Views and PL/SQL

Encrypted Format Login

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

Informix BYTE and TEXT Data Types

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

- An unsupported SQL function cannot be used in a SQL statement that accesses a column defined as Informix data type TEXT.

- You cannot use SQL*Plus to select data from a column defined as Informix data type TEXT when the data is greater than 80 characters in length. Oracle Corporation recommends using Pro*C or Oracle Call Interface to access such data in a Informix database.

- BYTE and TEXT data types must be NULLABLE for INSERT or UPDATE to work.

- A table including a BYTE or TEXT column must have a unique index defined on the table or the table must have a separate column that serves as a primary key.

- BYTE and TEXT data in a view cannot be accessed.

- BYTE and TEXT data cannot be read through pass-through queries.

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

> **See Also:** Appendix B, "Supported SQL Syntax and Functions".

Schema Names and PL/SQL

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

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

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

Data Dictionary Views and PL/SQL

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

4

Case Studies

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

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

This chapter contains the following sections:

- Case Descriptions
- CD-ROM Contents
- Demonstration Files
- Demonstration Requirements
- Creating Demonstration Tables
- Case 1: Simple Queries
- Case 2: A More Complex Query
- Case 3: Joining Informix Tables
- Case 4: Write Capabilities
- Case 5: Data Dictionary Query
- Case 6: The Pass-Through Feature

Case Descriptions

The cases illustrate:

- A simple query (Case 1)
- A more complex query (Case 2)
- Joining Informix tables (Case 3)
- Write capabilities (Case 4)
- A data dictionary query (Case 5)
- The pass-through feature (Case 6)

CD-ROM Contents

The distribution CD-ROM contains the following:

- Demonstration files

- One SQL script file that creates the demonstration tables in the Informix database

- One SQL script file that drops the demonstration tables from the Informix database

Demonstration Files

After a successful gateway installation, use the demonstration files stored in the directory $ORACLE_HOME/tg4ifmx/demo where $ORACLE_HOME is the $ORACLE_HOME directory under which the gateway is installed. The directory contains the following demonstration files:

Demonstration Files	Demonstration Files
bldifmx.sql	case4c.sql
case1.sql	case5.sql
case2.sql	case6a.sql
case3.sql	case6b.sql
case4a.sql	case7.sql
case4b.sql	dropifmx.sql

Demonstration Requirements

The case studies assume these requirements have been met:

- The gateway demonstration tables are installed in the Informix database

- The Oracle server has an account named SCOTT with a password of TIGER

- The Oracle server has a database link called GTWLINK (set up as public or private to the user SCOTT) which connects the gateway to a Informix database as SCOTT with password TIGER2

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

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

- Oracle Net Services is configured correctly and running.

- The Informix environment variable, INFORMIXDIR, is set correctly.

Creating Demonstration Tables

The case studies are based on the GTW_EMP, GTW_DEPT, and GTW_SALGRADE tables. If the demonstration tables have not been created in the Informix database, use the bldifmx.sql script to create them, as follows:

Set environment variable DELIMIDENT.

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

```
$ DELIMIDENT = y; export DELIMIDENT
```

If you have the C Shell, enter the following:

```
$ setenv DELIMIDENT y
```

```
$ cd $ORACLE_HOME/tg4ifmx/demo
$ dbaccess database_name bldifmx.sql
```

The script creates the demonstration tables in the Informix database accordingly:

```
CREATE TABLE GTW_EMP (
EMPNO      SMALLINT NOT NULL
ENAME      VARCHAR(10),
JOB        VARCHAR(9),
MGR        SMALLINT,
HIREDATE   DATETIME,
SAL        NUMERIC(7,2),
COMM       NUMERIC(7,2),
DEPTNO     SMALLINT)
CREATE TABLE GTW_DEPT (
DEPTNO     SMALLINT NOT NULL,
DNAME      VARCHAR(14),
LOC        VARCHAR(13))
CREATE TABLE GTW_SALGRADE (
GRADE      MONEY,
LOSAL      NUMERIC(9,4),
HISAL      NUMERIC(9,4))
```

Demonstration Table Definitions

The following table definitions use information retrieved by the SQL*PLUS DESCRIBE command:

GTW_EMP

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

GTW_DEPT

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

GTW_SALGRADE

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

Demonstration Table Contents

The contents of the Informix tables are:

GTW_EMP

```
EMPNO ENAME   JOB         MGR   HIREDATE   SAL   COMM   DEPTNO
```

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

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

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

Case 1: Simple Queries

Case 1 demonstrates the following:

- A simple query.

- A simple query retrieving full date information.

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

Case 2: A More Complex Query

Case 2 demonstrates the following:

- The functions SUM(*expression*) and NVL(*expr1, expr2*) in the SELECT list.

- The GROUP BY and HAVING clauses.

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

Case 3: Joining Informix Tables

Case 3 demonstrates the following:

- Joins between Informix tables.

- Subselects.

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

Case 4: Write Capabilities

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

- DELETE Statement

- UPDATE Statement

- INSERT Statement

DELETE Statement

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

UPDATE Statement

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

INSERT Statement

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

Case 5: Data Dictionary Query

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

Case 6: The Pass-Through Feature

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

This case demonstrates:

- A pass-through UPDATE statement using bind variables

- A pass-through SELECT statement

UPDATE Statement

Case 6a provides an example of a pass-through UPDATE statement with bind variables. In this example, the salary for EMPNO 7934 is set to 4000.

SELECT Statement

Case 6b provides an example of a pass-through SELECT statement. The data that is returned from the SELECT statement is inserted into a local table at the Oracle database server.

A

Data Type Conversion

This appendix contains the following section:

- Data Type Conversion

Data Type Conversion

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

Table A–1   Data Type Conversions

Informix	Oracle	Comment
BYTE	LONG RAW	-
CHAR	CHAR	-
DATE	DATE	-
DATETIME	DATE	-
DECIMAL	FLOAT(49)	-
DOUBLE PRECISION	FLOAT(49)	-
FLOAT	FLOAT(49)	-
INTEGER	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
INTERVAL	VARCHAR2(17)	-
MONEY	NUMBER(p[,s])	-
NCHAR	Not supported	-
NUMERIC	NUMBER(p[,s])	-
NVARCHAR	Not supported	-
REAL	FLOAT(23)	Precision is 6
SERIAL	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
SMALLFLOAT	FLOAT(23)	Precision is 6
SMALLINT	NUMBER(5)	NUMBER range is -32,767 to 32,767
TEXT	LONG	-
VARCHAR	VARCHAR2	If a length is not specified as part of VARCHAR, the data type is converted to VARCHAR2(1)

In addition to the rules shown in the preceding table, if the maximum size for an Informix data type is smaller or larger than the corresponding Oracle data type, data

might be lost. For example, if an Oracle table is defined with a column of VARCHAR2(300) and you use the COPY statement to copy the Oracle table to the Informix table where the Informix column is defined as VARCHAR(255), the data might be truncated.

Supported SQL Syntax and Functions

This appendix contains the following sections:

- Supported SQL Statements
- Oracle Functions

Supported SQL Statements

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

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

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

DELETE

The DELETE statement is fully supported. However, only Oracle functions supported by Informix can be used. Also, you cannot have SQL statements in the subquery that refer to the same table name in the outer query.

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

INSERT

The INSERT statement is fully supported. However, only Oracle functions supported by Informix can be used. Also, you cannot have SQL statements in the subquery that refer to the same table name in the outer query.

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

SELECT

The SELECT statement is fully supported, with these exceptions:

- CONNECT BY *condition*
- NOWAIT
- START WITH *condition*

■ Subquery in HAVING clause

UPDATE

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

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

Oracle Functions

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

Functions Not Supported by Informix

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

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

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

Functions Supported by Informix

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

- Arithmetic Operators
- Comparison Operators
- Group Functions
- String Functions
- Pattern Matches
- Date Functions
- Other Functions

Arithmetic Operators

Oracle	Informix
+	+
-	-
*	*
/	/

Comparison Operators

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

Group Functions

Oracle	Informix
AVG	AVG
COUNT	COUNT
MAX	MAX
MIN	MIN
SUM	SUM

String Functions

Oracle	Informix
||, CONCAT	||
ASCII	ASCII
CHR	CHR
LENGTH	LENGTH

Pattern Matches

Oracle	Informix
LIKE 'a%'	LIKE "a%"
LIKE 'a_'	LIKE "a_"
LIKE 'a\%' ESCAPE '\'	LIKE "a\%" ESCAPE "\"
NOT LIKE	NOT LIKE

Date Functions

Oracle	Informix
date + number	*date + number*
date - number	*date - number*
date + date	*date + date*
date - date	*date - date*

Other Functions

Oracle	Informix
ABS	ABS
COS	COS
EXP	EXP
LOG10	LOG10
LN	LOGN
LTRIM(*char*)	TRIM(LEADING FROM *char*)
MOD	MOD
POWER (*m,n*)	POW(*m,n*)
RTRIM(*char*)	TRIM(TRAILING FROM *char*)
ROUND (with 1 argument)	ROUND
SIN	SIN
SQRT	SQRT
TAN	TAN
TRUNC (with 1 argument)	TRUNC

Functions Supported by the Gateway

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

The gateway, however, does support one function even though there is no equivalent in Informix. This function is the TO_DATE function:

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

date_string is converted to a string with the following format:

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

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

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

C

Data Dictionary

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

This appendix contains the following sections:

- Data Dictionary Support
- Data Dictionary Mapping
- Gateway Data Dictionary Descriptions

Data Dictionary Support

The following paragraphs describe the data dictionary support of the Oracle Transparent Gateway for Informix.

Informix System Catalog Tables

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

Accessing the Gateway Data Dictionary

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

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

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

1. Maps the requested table, view, or synonym to one or more Informix system catalog table names. The gateway translates all data dictionary column names to their corresponding Informix column names within the query. If the mapping

involves one Informix system catalog table, the gateway translates the requested table name to its corresponding Informix system catalog table name within the query. If the mapping involves multiple Informix system catalog tables, the gateway constructs a join in the query using the translated Informix system catalog table names.

2. Sends the translated query to Informix.

3. Might convert the retrieved Informix data to give it the appearance of the Oracle database server data dictionary table.

4. Passes the data dictionary information from the translated Informix system catalog table to the Oracle database server.

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

Direct Queries to Informix Tables

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

Supported Views and Tables

The gateway supports the following views and tables:

Supported Views and Tables	Supported Views and Tables
ALL_CATALOG	ALL_COL_COMMENTS
ALL_COL_PRIVS	ALL_CONS_COLUMNS
ALL_CONSTRAINTS	ALL_IND_COLUMNS
ALL_INDEXES	ALL_OBJECTS
ALL_SYNONYMS	ALL_TAB_COLUMNS
ALL_TAB_COMMENTS	ALL_TAB_PRIVS
ALL_TABLES	ALL_USERS
ALL_VIEWS	COLUMN_PRIVILEGES
DBA_CATALOG	DBA_COL_COMMENTS
DBA_OBJECTS	DBA_TABLES
DBA_TAB_COLUMNS	DBA_TAB_COMMENTS
DICT_COLUMNS	DICTIONARY
DUAL	TABLE_PRIVILEGES
USER_CATALOG	USER_COL_COMMENTS
USER_COL_PRIVS	USER_CONS_COLUMNS
USER_CONSTRAINTS	USER_IND_COLUMNS
USER_INDEXES	USER_OBJECTS

Supported Views and Tables	Supported Views and Tables
USER_SYNONYMS	USER_TAB_COLUMNS
USER_TAB_COMMENTS	USER_TAB_PRIVS
USER_TABLES	USER_USERS
USER_VIEWS	

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

Queries through the gateway of any data dictionary table or view beginning with ALL_ can returns rows from the Informix database even when access privileges for those Informix objects have not been granted. When querying an Oracle database with the Oracle data dictionary, rows are returned only for those objects you are permitted to access.

Data Dictionary Mapping

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

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

View Name	Informix System Catalog Table Name
ALL_CATALOG	systables
ALL_COL_COMMENTS	systables +syscolumns
ALL_COL_PRIVS	systables + syscolumns + syscolauth
ALL_CONS_COLUMNS	systables + sysconstraints + syscolumns + sysindexes
ALL_CONSTRAINTS	systables + sysconstraints + sysreferences
ALL_IND_COLUMNS	systables + sysindexes + syscolumns
ALL_INDEXES	sysindexes + systables
ALL_OBJECTS	systables + sysindexes + sysprocedures + sysprocplan
ALL_SYNONYMS	systables + syssynonyms + syssyntable
ALL_TAB_COLUMNS	systables + syscolumns
ALL_TAB_COMMENTS	systables
ALL_TAB_PRIVS	systables + systabauth
ALL_TABLES	systables
ALL_USERS	sysusers
ALL_VIEWS	systables + sysviews
COLUMN_PRIVILEGES	systables + syscolauth + syscolumns
DBA_CATALOG	systables
DBA_COL_COMMENTS	systables + syscolumns

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

View Name	Informix System Catalog Table Name
DBA_OBJECTS	systables + sysindexes + sysprocedures + sysprocplan
DBA_TABLES	systables
DBA_TAB_COLUMNS	systables + syscolumns
DBA_TAB_COMMENTS	systables
DICT_COLUMNS	systables + syscolumns
DICTIONARY	systables
DUAL	*(Defined in the Gateway)*
TABLE_PRIVILEGES	systabauth + systables
USER_CATALOG	systables
USER_COL_COMMENTS	systables + syscolumns
USER_COL_PRIVS	systables + syscolumns + syscolauth
USER_CONS_COLUMNS	systables + sysconstraints + syscolumns + sysindexes
USER_CONSTRAINTS	systables + sysconstraints + sysreferences
USER_IND_COLUMNS	systables + sysindexes + syscolumns
USER_INDEXES	systables + sysindexes
USER_OBJECTS	systables + sysindexes + sysprocedures + sysprocplan
USER_SYNONYMS	systables + syssynonyms + syssyntable
USER_TAB_COLUMNS	systables + syscolumns
USER_TAB_COMMENTS	systables
USER_TAB_PRIVS	systables + systabauth
USER_TABLES	systables
USER_USERS	sysusers
USER_VIEWS	systables + sysviews

Default Column Values

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

Gateway Data Dictionary Descriptions

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

- Name, data type, and width of each column

- The contents of columns with fixed values

They are described here with information retrieved by an Oracle SQL*Plus DESCRIBE command. The values in the Null? column might differ from the Oracle database

server data dictionary tables and views. Any default value is shown to the right of an item, but this is not information returned by DESCRIBE.

Table C–2   ALL_CATALOG

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(7)	"TABLE" or "VIEW" or "SYNONYM"

Table C–3   ALL_COL_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–4   ALL_COL_PRIVS

Name	Null?	Type	Value
GRANTOR	-	VARCHAR2(32)	-
GRANTEE	-	VARCHAR2(32)	-
TABLE_SCHEMA	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
PRIVILEGE	-	VARCHAR2(10)	"SELECT" or "UPDATE" or "REFERENCES"
GRANTABLE	-	VARCHAR2(3)	"YES″ or "NO"

Table C–5   ALL_CONS_COLUMNS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
CONSTRAINT_NAME	-	VARCHAR2(128)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
POSITION	NOT NULL	NUMBER(10)	0

Table C–6   ALL_CONSTRAINTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(32)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2(128)	-

Table C–6 (Cont.) ALL_CONSTRAINTS

Name	Null?	Type	Value
CONSTRAINT_TYPE	NOT NULL	VARCHAR2(1)	"R" or "P" or "U" or "C"
TABLE_NAME	NOT NULL	VARCHAR2(128)	-
SEARCH_CONDITION	NOT NULL	CHAR(1)	" "
R_OWNER	NOT NULL	CHAR(1)	" "
R_CONSTRAINT_NAME	NOT NULL	CHAR(1)	" "
DELETE_RULE	NOT NULL	CHAR(1)	" "
STATUS	NOT NULL	CHAR(1)	" "
DEFERRABLE	NOT NULL	CHAR(1)	" "
DEFERRED	NOT NULL	CHAR(1)	" "
VALIDATED	NOT NULL	CHAR(1)	" "
GENERATED	NOT NULL	CHAR(1)	" "
BAD	NOT NULL	CHAR(1)	" "
RELY	NOT NULL	CHAR(1)	" "
LAST_CHANGE	-	DATE	-

Table C–7 ALL_IND_COLUMNS

Name	Null?	Type	Value
INDEX_OWNER	-	VARCHAR2(32)	-
INDEX_NAME	-	VARCHAR2(128)	-
TABLE_OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
COLUMN_POSITION	NOT NULL	NUMBER(10)	0
COLUMN_LENGTH	NOT NULL	NUMBER(10)	0
DESCEND	NOT NULL	CHAR(1)	" "

Table C–8 ALL_INDEXES

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
INDEX_NAME	-	VARCHAR2(128)	-
INDEX_TYPE	-	VARCHAR2(1)	NULL
TABLE_OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR(5)	"TABLE"
UNIQUENESS	-	VARCHAR2(9)	"UNIQUE" or "NONUNIQUE"
COMPRESSION	-	VARCHAR2(1)	NULL

Table C–8   (Cont.) ALL_INDEXES

Name	Null?	Type	Value
PREFIX_LENGTH	-	NUMBER	0
TABLESPACE_NAME	-	VARCHAR2(1)	NULL
INI_TRANS	-	NUMBER	0
MAX_TRANS	-	NUMBER	0
INITIAL_EXTENT	-	NUMBER	0
NEXT_EXTENT	-	NUMBER	0
MIN_EXTENTS	-	NUMBER	0
MAX_EXTENTS	-	NUMBER	0
PCT_INCREASE	-	NUMBER	0
PCT_THRESHOLD	-	NUMBER	0
INCLUDE_COLUMN	-	NUMBER	0
FREELISTS	-	NUMBER	0
FREELIST_GROUPS	-	NUMBER	0
PCT_FREE	-	NUMBER	0
LOGGING	-	VARCHAR2(1)	NULL
BLEVEL	-	NUMBER	0
LEAF_BLOCKS	-	NUMBER	0
DISTINCT_KEYS	-	NUMBER	0
AVG_LEAF_BLOCKS_PER_KEY	-	NUMBER	0
AVG_DATA_BLOCKS_PER_KEY	-	NUMBER	0
CLUSTERING_FACTOR	-	NUMBER	0
STATUS	-	VARCHAR2(1)	NULL
NUM_ROWS	-	NUMBER	0
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
DEGREE	-	VARCHAR2(1)	NULL
INSTANCES	-	VARCHAR2(1)	NULL
PARTITIONED	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL
BUFFER_POOL	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
DURATION	-	VARCHAR2(1)	NULL
PCT_DIRECT_ACCESS	-	NUMBER	0
ITYP_OWNER	-	VARCHAR2(1)	NULL

Table C–8   (Cont.) ALL_INDEXES

Name	Null?	Type	Value
ITYP_NAME	-	VARCHAR2(1)	NULL
PARAMETERS	-	VARCHAR2(1)	NULL
GLOBAL_STATS	-	VARCHAR2(1)	NULL
DOMIDX_STATUS	-	VARCHAR2(1)	NULL
DOMIDX_OPSTATUS	-	VARCHAR2(1)	NULL
FUNCIDX_STATUS	-	VARCHAR2(1)	NULL

Table C–9   ALL_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
OBJECT_NAME	-	VARCHAR2(128)	-
SUBOBJECT_NAME	-	VARCHAR2(1)	NULL
OBJECT_ID	-	NUMBER	-
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2(9)	"TABLE" or "VIEW" or "SYNONYM" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	-
LAST_DDL_TIME	-	DATE	-
TIMESTAMP	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL

Table C–10   ALL_SYNONYMS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(32)	-
SYNONYM_NAME	NOT NULL	VARCHAR2(128)	-
TABLE_OWNER	NOT NULL	VARCHAR2(32)	-
TABLE_NAME	NOT NULL	VARCHAR2(128)	-
DB_LINK	-	CHAR(1)	NULL

Table C–11   ALL_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-

Table C–11   (Cont.) ALL_TAB_COLUMNS

Name	Null?	Type	Value
COLUMN_NAME	-	VARCHAR2(128)	-
DATA_TYPE	-	VARCHAR2(8)	-
DATA_TYPE_MOD	-	VARCHAR2(1)	NULL
DATA_TYPE_OWNER	-	VARCHAR2(1)	NULL
DATA_LENGTH	-	NUMBER	-
DATA_PRECISION	-	NUMBER	-
DATA_SCALE	-	NUMBER	-
NULLABLE	-	VARCHAR2(1)	"Y" or "N"
COLUMN_ID	-	NUMBER(5)	-
DEFAULT_LENGTH	-	NUMBER	0
DATA_DEFAULT	-	VARCHAR2(1)	NULL
NUM_DISTINCT	-	NUMBER	0
LOW_VALUE	-	NUMBER	0
HIGH_VALUE	-	NUMBER	0
DENSITY	-	NUMBER	0
NUM_NULLS	-	NUMBER	0
NUM_BUCKETS	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
SAMPLE_SIZE	-	NUMBER	0
CHARACTER_SET_NAME	-	VARCHAR2(1)	NULL
CHAR_COL_DECL_LENGTH	-	NUMBER	0
GLOBAL_STATS	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
AVG_COL_LEN	-	NUMBER	0

Table C–12   ALL_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS	-	VARCHAR2(1)	NULL

Table C–13   ALL_TAB_PRIVS

Name	Null?	Type	Value
GRANTOR	-	VARCHAR2(32)	-

Table C–13   (Cont.) ALL_TAB_PRIVS

Name	Null?	Type	Value
GRANTEE	-	VARCHAR2(32)	-
TABLE_SCHEMA	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
PRIVILEGE	-	VARCHAR2(10)	"SELECT" or "UPDATE" or "INSERT" or "DELETE" or "INDEX" or "ALTER" or "REFERENCES"
GRANTABLE	-	VARCHAR2 (3)	"YES"

Table C–14   ALL_TABLES

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLESPACE_NAME	-	VARCHAR2(1)	NULL
CLUSTER_NAME	-	VARCHAR2(1)	NULL
IOT_NAME	-	VARCHAR2(1)	NULL
PCT_FREE	-	NUMBER	0
PCT_USED	-	NUMBER	0
INI_TRANS	-	NUMBER	0
MAX_TRANS	-	NUMBER	0
INITIAL_EXTENT	-	NUMBER	0
NEXT_EXTENT	-	NUMBER	0
MIN_EXTENTS	-	NUMBER	0
MAX_EXTENTS	-	NUMBER	0
PCT_INCREASE	-	NUMBER	0
FREELISTS	-	NUMBER	0
FREELIST_GROUPS	-	NUMBER	0
LOGGING	-	VARCHAR2(1)	NULL
BACKED_UP	-	VARCHAR2(1)	NULL
NUM_ROWS	-	NUMBER(10)	-
BLOCKS	-	NUMBER	0
EMPTY_BLOCKS	-	NUMBER	0
AVG_SPACE	-	NUMBER	0
CHAIN_CNT	-	NUMBER	0
AVG_ROW_LEN	-	NUMBER	0
AVG_SPACE_FREELIST_BLOCKS	-	NUMBER	0

Table C–14   (Cont.) ALL_TABLES

Name	Null?	Type	Value
NUM_FREELIST_BLOCKS	-	NUMBER	0
DEGREE	-	VARCHAR2(1)	NULL
INSTANCES	-	VARCHAR2(1)	NULL
CACHE	-	VARCHAR2(1)	NULL
TABLE_LOCK	-	VARCHAR2(1)	NULL
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
PARTITIONED	-	VARCHAR2(1)	NULL
IOT_TYPE	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARHCAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL
NESTED	-	VARCHAR2(1)	NULL
BUFFER_POOL	-	VARCHAR2(1)	NULL
ROW_MOVEMENT	-	VARCHAR2(1)	NULL
GLOBAL_STATS	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
DURATION	-	VARHCAR2(1)	NULL
SKIP_CORRUPT	-	VARCHAR2(1)	NULL
MONITORING	-	VARCHAR2(1)	NULL

Table C–15   ALL_USERS

Name	Null?	Type	Value
USERNAME	NOT NULL	VARCHAR2(32)	-
USER_ID	NOT NULL	NUMBER	0
CREATED	NOT NULL	DATE	SYSDATE

Table C–16   ALL_VIEWS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
VIEW_NAME	-	VARCHAR2(128)	-
TEXT_LENGTH	NOT NULL	NUMBER(10)	64
TEXT	-	VARCHAR2(64)	-
TYPE_TEXT_LENGTH	NOT NULL	NUMBER(10)	0
TYPE_TEXT	NOT NULL	CHAR(1)	" "
OID_TEXT_LENGTH	NOT NULL	NUMBER(10)	0
OID_TEXT	NOT NULL	CHAR(1)	" "

Table C–16   (Cont.) ALL_VIEWS

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

Table C–17   COLUMN_PRIVILEGES

Name	Null?	Type	Value
GRANTEE	-	VARCHAR2(32)	-
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
GRANTOR	-	VARCHAR2(32)	-
INSERT_PRIV	-	VARCHAR2(1)	"Y"
UPDATE_PRIV	-	VARCHAR2(1)	"Y"
REFERENCES_PRIV	-	VARCHAR2(1)	NULL
CREATED	-	DATE	SYSDATE

Table C–18   DBA_CATALOG

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(7)	"TABLE" or "VIEW" or "SYNONYM"

Table C–19   DBA_COL_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–20   DBA_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
OBJECT_NAME	-	VARCHAR2(128)	-
SUBOBJECT_NAME	-	VARCHAR2(1)	NULL
OBJECT_ID	-	NUMBER	-
DATA_OBJECT_ID	-	NUMBER	0

Table C–20   (Cont.) DBA_OBJECTS

Name	Null?	Type	Value
OBJECT_TYPE	-	VARCHAR2(9)	"TABLE" or "VIEW" or "SYNONYM" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	-
LAST_DDL_TIME	-	DATE	-
TIMESTAMP	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL

Table C–21   DBA_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
DATA_TYPE	-	VARCHAR2(8)	-
DATA_TYPE_MOD	-	VARCHAR2(1)	NULL
DATA_TYPE_OWNER	-	VARCHAR2(1)	NULL
DATA_LENGTH	-	NUMBER	-
DATA_PRECISION	-	NUMBER	-
DATA_SCALE	-	NUMBER	-
NULLABLE	-	VARCHAR2(1)	"Y" or "N"
COLUMN_ID	-	NUMBER(5)	-
DEFAULT_LENGTH	-	NUMBER	0
DATA_DEFAULT	-	VARCHAR2(1)	NULL
NUM_DISTINCT	-	NUMBER	0
LOW_VALUE	-	NUMBER	0
HIGH_VALUE	-	NUMBER	0
DENSITY	-	NUMBER	0
NUM_NULLS	-	NUMBER	0
NUM_BUCKETS	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
SAMPLE_SIZE	-	NUMBER	0
CHARACTER_SET_NAME	-	VARCHAR2(1)	NULL

Table C–21   (Cont.) DBA_TAB_COLUMNS

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

Table C–22   DBA_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS	-	VARCHAR2(1)	NULL

Table C–23   DBA_TABLES

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLESPACE_NAME	-	VARCHAR2(1)	NULL
CLUSTER_NAME	-	VARCHAR2(1)	NULL
IOT_NAME	-	VARCHAR2(1)	NULL
PCT_FREE	-	NUMBER	0
PCT_USED	-	NUMBER	0
INI_TRANS	-	NUMBER	0
MAX_TRANS	-	NUMBER	0
INITIAL_EXTENT	-	NUMBER	0
NEXT_EXTENT	-	NUMBER	0
MIN_EXTENTS	-	NUMBER	0
MAX_EXTENTS	-	NUMBER	0
PCT_INCREASE	-	NUMBER	0
FREELISTS	-	NUMBER	0
FREELIST_GROUPS	-	NUMBER	0
LOGGING	-	VARCHAR2(1)	NULL
BACKED_UP	-	VARCHAR2(1)	NULL
NUM_ROWS	-	NUMBER(10)	
BLOCKS	-	NUMBER	0
EMPTY_BLOCKS	-	NUMBER	0
AVG_SPACE	-	NUMBER	0

Table C–23   (Cont.) DBA_TABLES

Name	Null?	Type	Value
CHAIN_CNT	-	NUMBER	0
AVG_ROW_LEN	-	NUMBER	0
AVG_SPACE_FREELIST_BLOCKS	-	NUMBER	0
NUM_FREELIST_BLOCKS	-	NUMBER	0
DEGREE	-	VARCHAR2(1)	NULL
INSTANCES	-	VARCHAR2(1)	NULL
CACHE	-	VARCHAR2(1)	NULL
TABLE_LOCK	-	VARCHAR2(1)	NULL
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
PARTITIONED	-	VARCHAR2(1)	NULL
IOT_TYPE	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARHCAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL
NESTED	-	VARCHAR2(1)	NULL
BUFFER_POOL	-	VARCHAR2(1)	NULL
ROW_MOVEMENT	-	VARCHAR2(1)	NULL
GLOBAL_STATS	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
DURATION	-	VARHCAR2(1)	NULL
SKIP_CORRUPT	-	VARCHAR2(1)	NULL
MONITORING	-	VARCHAR2(1)	NULL

Table C–24   DICT_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
COMMENTS	NOT NULL	VARCHAR2(1)	-

Table C–25   DICTIONARY

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–26   DUAL

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

Table C–27   TABLE_PRIVILEGES

Name	Null?	Type	Value
GRANTEE	-	VARCHAR2(32)	-
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
GRANTOR	-	VARCHAR2(32)	-
SELECT_PRIV	-	VARCHAR2(1)	"Y" or "N"
INSERT_PRIV	-	VARCHAR2(1)	"Y" or "N"
DELETE_PRIV	-	VARCHAR2(1)	"Y" or "N"
UPDATE_PRIV	-	VARCHAR2(1)	"Y" or "N"
REFERENCES_PRIV	-	VARCHAR2(1)	"Y" or "N"
ALTER_PRIV	-	VARCHAR2(1)	"Y" or "N"
INDEX_PRIV	-	VARCHAR2(1)	"Y" or "N"
CREATED	-	DATE	SYSDATE

Table C–28   USER_CATALOG

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(7)	"TABLE" or "VIEW" or "SYNONYM"

Table C–29   USER_COL_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
COMMENTS	-	VARCHAR2(1)	NULL

Table C–30   USER_COL_PRIVS

Name	Null?	Type	Value
GRANTOR	-	VARCHAR2 (32)	-
OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
GRANTEE	-	VARCHAR2(32)	-

Table C–30   (Cont.) USER_COL_PRIVS

Name	Null?	Type	Value
PRIVILEGE	-	VARCHAR2(10)	"SELECT" or "UPDATE" or "REFERENCES"
GRANTABLE	-	VARCHAR2(3)	"YES" or "NO"

Table C–31   USER_CONS_COLUMNS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
CONSTRAINT_NAME	-	VARCHAR2(128)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
POSITION	-	NUMBER	0

Table C–32   USER_CONSTRAINTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2(32)	-
CONSTRAINT_NAME	-	VARCHAR2(128)	-
CONSTRAINT_TYPE	-	VARCHAR2(1)	"R" or "P" or "U" or "C"
TABLE_NAME	-	VARCHAR2(128)	-
SEARCH_CONDITION	-	VARCHAR2(1)	NULL
R_OWNER	-	VARCHAR2(32)	NULL
R_CONSTRAINT_NAME	-	VARCHAR2(128)	NULL
DELETE_RULE	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
DEFERRABLE	-	VARCHAR2(1)	NULL
DEFERRED	-	VARCHAR2(1)	NULL
VALIDATED	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
BAD	-	VARCHAR2(1)	NULL
RELY	-	VARCHAR2(1)	NULL
LAST_CHANGE	-	DATE	

Table C–33   USER_IND_COLUMNS

Name	Null?	Type	Value
INDEX_NAME	-	VARCHAR2(128)	-
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-

Table C–33   (Cont.) USER_IND_COLUMNS

Name	Null?	Type	Value
COLUMN_POSITION	-	NUMBER	0
COLUMN_LENGTH	-	NUMBER	0
DESCEND	-	VARCHAR2(1)	-

Table C–34   USER_INDEXES

Name	Null?	Type	Value
INDEX_NAME	-	VARCHAR2(128)	-
INDEX_TYPE	-	VARCHAR2(1)	NULL
TABLE_OWNER	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE"
UNIQUENESS	-	VARCHAR2(9)	"UNIQUE" or "NONUNIQUE"
COMPRESSION	-	VARCHAR2(1)	NULL
PREFIX_LENGTH	-	NUMBER	0
TABLESPACE_NAME	-	VARCHAR2(1)	NULL
INI_TRANS	-	NUMBER	0
MAX_TRANS	-	NUMBER	0
INITIAL_EXTENT	-	NUMBER	0
NEXT_EXTENT	-	NUMBER	0
MIN_EXTENTS	-	NUMBER	0
MAX_EXTENTS	-	NUMBER	0
PCT_INCREASE	-	NUMBER	0
PCT_THRESHOLD	-	NUMBER	0
INCLUDE_COLUMN	-	NUMBER	0
FREELISTS	-	NUMBER	0
FREELIST_GROUPS	-	NUMBER	0
PCT_FREE	-	NUMBER	0
LOGGING	-	VARCHAR2(1)	NULL
BLEVEL	-	NUMBER	0
LEAF_BLOCKS	-	NUMBER	0
DISTINCT_KEYS	-	NUMBER	-
AVG_LEAF_BLOCKS_PER_KEY	-	NUMBER	0
AVG_DATA_BLOCKS_PER_KEY	-	NUMBER	0
CLUSTERING_FACTOR	-	NUMBER	0
STATUS	-	VARCHAR2(1)	NULL
NUM_ROWS	-	NUMBER	0

Table C–34   (Cont.) USER_INDEXES

Name	Null?	Type	Value
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
DEGREE	-	VARCHAR2(1)	NULL
INSTANCES	-	VARCHAR2(1)	NULL
PARTITIONED	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL
BUFFER_POOL	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
DURATION	-	VARHCAR2(1)	NULL
PCT_DIRECT_ACCESS	-	NUMBER	0
ITYP_OWNER	-	VARCHAR2(1)	NULL
ITYP_NAME	-	VARCHAR2(1)	NULL
PARAMETERS	-	VARCHAR2(1)	NULL
GLOBAL_STATS	-	VARCHAR2(1)	NULL
DOMIDX_STATUS	-	VARCHAR2(1)	NULL
DOMIDX_OPSTATUS	-	VARCHAR2(1)	NULL
FUNCIDX_STATUS	-	VARCHAR2(1)	NULL

Table C–35   USER_OBJECTS

Name	Null?	Type	Value
OBJECT_NAME	-	VARCHAR2(128)	-
SUBOBJECT_NAME	-	VARCHAR2(1)	NULL
OBJECT_ID	-	NUMBER	-
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2(9)	"TABLE" or "VIEW" or "SYNONYM" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	-
LAST_DDL_TIME	-	DATE	-
TIMESTAMP	-	VARCHAR2(1)	NULL
STATUS	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARCHAR2(1)	NULL
GENERATED	-	VARCHAR2(1)	NULL

Table C–35   (Cont.) USER_OBJECTS

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

Table C–36   USER_SYNONYMS

Name	Null?	Type	Value
SYNONYM_NAME	-	VARCHAR2(128)	-
TABLE_OWNER'	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
DB_LINK	-	VARCHAR2(1)	NULL

Table C–37   USER_TAB_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
COLUMN_NAME	-	VARCHAR2(128)	-
DATA_TYPE	-	VARCHAR2(8)	-
DATA_TYPE_MOD	-	VARCHAR2(1)	NULL
DATA_TYPE_OWNER	-	VARCHAR2(1)	NULL
DATA_LENGTH	-	NUMBER	-
DATA_PRECISION	-	NUMBER	-
DATA_SCALE	-	NUMBER	-
NULLABLE	-	VARCHAR2(1)	"Y" or "N"
COLUMN_ID	-	NUMBER(5)	-
DEFAULT_LENGTH	-	NUMBER	0
DATA_DEFAULT	-	VARCHAR2(1)	NULL
NUM_DISTINCT	-	NUMBER	0
LOW_VALUE	-	NUMBER	0
HIGH_VALUE	-	NUMBER	0
DENSITY	-	NUMBER	0
NUM_NULLS	-	NUMBER	0
NUM_BUCKETS	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
SAMPLE_SIZE	-	NUMBER	0
CHARACTER_SET_NAME	-	VARCHAR2(1)	NULL
CHAR_COL_DECL_LENGTH	-	NUMBER	0
GLOBAL_STATS	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
AVG_COL_LEN	-	NUMBER	0

Table C–38   USER_TAB_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
TABLE_TYPE	-	VARCHAR2(5)	"TABLE" or "VIEW"
COMMENTS	-	VARCHAR2(1)	NULL

Table C–39   USER_TAB_PRIVS

Name	Null?	Type	Value
GRANTEE	-	VARCHAR2(32)	-
TABLE_SCHEMA	-	VARCHAR2(32)	-
TABLE_NAME	-	VARCHAR2(128)	-
GRANTOR	-	VARCHAR2(32)	-
PRIVILEGE	-	VARCHAR2(6)	"SELECT" or "UPDATE" or "INSERT" or "DELETE" or "INDEX" or " "
GRANTABLE	-	VARCHAR2(3)	"YES"

Table C–40   USER_TABLES

Name	Null?	Type	Value
TABLE_NAME	-	VARCHAR2(128)	-
TABLESPACE_NAME	-	VARCHAR2(1)	NULL
CLUSTER_NAME	-	VARCHAR2(1)	NULL
IOT_NAME	-	VARCHAR2(1)	NULL
PCT_FREE	-	NUMBER	0
PCT_USED	-	NUMBER	0
INI_TRANS	-	NUMBER	0
MAX_TRANS	-	NUMBER	0
INITIAL_EXTENT	-	NUMBER	0
NEXT_EXTENT	-	NUMBER	0
MIN_EXTENTS	-	NUMBER	0
MAX_EXTENTS	-	NUMBER	0
PCT_INCREASE	-	NUMBER	0
FREELISTS	-	NUMBER	0
FREELIST_GROUPS	-	NUMBER	0
LOGGING	-	VARCHAR2(1)	NULL
BACKED_UP	-	VARCHAR2(1)	NULL
NUM_ROWS	-	NUMBER(10)	0
BLOCKS	-	NUMBER	0

Table C–40   (Cont.) USER_TABLES

Name	Null?	Type	Value
EMPTY_BLOCKS	-	NUMBER	0
AVG_SPACE	-	NUMBER	0
CHAIN_CNT	-	NUMBER	0
AVG_ROW_LEN	-	NUMBER	0
AVG_SPACE_FREELIST_BLOCKS	-	NUMBER	0
NUM_FREELIST_BLOCKS	-	NUMBER	0
DEGREE	-	VARCHAR2(1)	NULL
INSTANCES	-	VARCHAR2(1)	NULL
CACHE	-	VARCHAR2(1)	NULL
TABLE_LOCK	-	VARCHAR2(1)	NULL
SAMPLE_SIZE	-	NUMBER	0
LAST_ANALYZED	-	DATE	to_date('01-01-1980', 'dd-mm-yyyy')
PARTITIONED	-	VARCHAR2(1)	NULL
IOT_TYPE	-	VARCHAR2(1)	NULL
TEMPORARY	-	VARHCAR2(1)	NULL
SECONDARY	-	VARCHAR2(1)	NULL
NESTED	-	VARCHAR2(1)	NULL
BUFFER_POOL	-	VARCHAR2(1)	NULL
ROW_MOVEMENT	-	VARCHAR2(1)	NULL
GLOBAL_STATS	-	VARCHAR2(1)	NULL
USER_STATS	-	VARCHAR2(1)	NULL
DURATION	-	VARCHAR2(1)	NULL
SKIP_CORRUPT	-	VARCHAR2(1)	NULL
MONITORING	-	VARCHAR2(1)	NULL

Table C–41   USER_USERS

Name	Null?	Type	Value
USERNAME	-	VARCHAR2(32)	-
USER_ID	-	NUMBER	-
ACCOUNT_STATUS	-	VARCHAR2(4)	"OPEN"
LOCK_DATE	-	DATE	NULL
EXPIRY_DATE	-	DATE	NULL
DEFAULT_TABLESPACE	-	VARCHAR2(1)	NULL
TEMPORARY_TABLESPACE	-	VARCHAR2(1)	NULL
CREATED	-	DATE	NULL
INITIAL_RSRC_CONSUMER_GROUP	-	VARCHAR2(1)	NULL

Table C–41   (Cont.) USER_USERS

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

Table C–42   USER_VIEWS

Name	Null?	Type	Value
VIEW_NAME	-	VARCHAR2(128)	-
TEXT_LENGTH	-	NUMBER	64
TEXT	-	VARCHAR2(64)	-
TYPE_TEXT_LENGTH	-	NUMBER	0
TYPE_TEXT	-	VARCHAR2(1)	NULL
OID_TEXT_LENGTH	-	NUMBER	0
OID_TEXT	-	VARCHAR2(1)	NULL
VIEW_TYPE_OWNER	-	VARCHAR2(1)	NULL
VIEW_TYPE	-	VARCHAR2(1)	NULL

D

Heterogeneous Services Initialization Parameters

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

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

- HS_FDS_CONNECT_INFO
- HS_FDS_PARSER_TOKEN_SIZE
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_TRANSACTION_MODEL

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

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

```
$ORACLE_HOME/tg4ifmx/admin/inittg4ifmx.ora
```

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

HS Initialization Parameter File Syntax

The syntax for the initialization parameter file is as follows:

1. The file is a sequence of commands.

2. Each command should start on a separate line.

3. End of line is considered a command terminator (unless escaped with a backslash).

4. Each command can have one of the following forms:

   a. `<param> = <value>`

b. `set <param> = <value>`

c. `private <param> = <value>`

d. `set private <param> = <value>`

Where:

`<param>` is an initialization parameter name.

`<value>` is the initialization parameter value.

`'set'` and `'private'` are keywords.

5. The keywords `'set'` and `'private'` are reserved. You cannot use either as an initialization parameter name. The `'set'` keyword indicates that the initialization parameter should be set as an environment variable in the agent. The `'private'` keyword indicates that the initialization parameter should be private to the agent and should not be uploaded to the server. Most initialization parameters should not be private. If, however, you are storing something sensitive like a password in the initialization parameter file, then you may not want it uploaded to the server because the initialization parameters and values are not encrypted when uploaded. Making these initialization parameters private prevents the upload from happening.

6. An initialization parameter name is a string of characters starting with a letter and consisting of letters, digits and underscores. Initialization parameter names are case sensitive.

7. An initialization parameter value is either:

   a. A string of characters that does not contain any backslashes, white space or double quotation marks (")

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

   * backslash (\) is the escape character

   * \n inserts a new line

   * \t inserts a tab

   * \" inserts a double quotation mark

   * \\ inserts a backslash

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

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

HS_FDS_CONNECT_INFO

Default Value	Range of Values
None	Not Applicable

Specifies the information needed to connect to the Informix database.

This is a required parameter, whose format is:

`HS_FDS_CONNECT_INFO=database_name@server_name`

Where:

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

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

If you specify only *database_name*, omitting *server_name*, the gateway binds to the database residing on the local Informix server (as specified in the environment variable INFORMIXSERVER).

HS_FDS_PARSER_TOKEN_SIZE

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

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

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

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

HS_FDS_RECOVERY_ACCOUNT

Default Value	Range of Values
RECOVER	Any valid userid

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

The name of the recovery account is case-sensitive.

HS_FDS_RECOVERY_PWD

Default Value	Range of Values
RECOVER	Any valid password

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

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

HS_FDS_TRACE_LEVEL

Default Value	Range of Values
OFF	OFF, ON

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

The following values are valid:

- OFF disables the tracing of error messages.

- ON enables the tracing of error messages that occur when you encounter problems. The results are written to a gateway connectivity log file, in $ORACLE_ HOME/TG4IFMX/LOG.

HS_FDS_TRANSACTION_LOG

Default Value	Range of Values
HS_TRANSACTION_LOG	Any valid table name

Specifies the name of the table created in the Informix database for logging transactions. For more information about the transaction model, see the HS_FDS_ TRANSACTION_MODEL parameter.

HS_FDS_TRANSACTION_MODEL

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

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

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the Informix 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 Informix database:

    - Transaction log table. The default table name is HS_TRANSACTION_LOG. A different name can be set using the HS_FDS_TRANSACTION_LOG parameter. The transaction log table must be granted SELECT, DELETE, and INSERT privileges set to public.

    - Recovery account. The account name is assigned with the HS_FDS_ RECOVERY_ACCOUNT parameter.

    - Recovery account password. The password is assigned with the HS_FDS_ RECOVERY_PWD parameter.

- READ_ONLY provides read access to the Informix database.

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

Index

VARCHAR2 data type, A-1