Oracle® Database Gateway for Informix

User's Guide

19c
F18234-01
April 2019

Oracle Database Gateway for Informix User's Guide, 19c

F18234-01

Copyright © 2002, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Rhonda Day

Contributing Authors: Vira Goorah, Juan Pablo Ahues-Vasquez, Peter Castro, Charles Benet, Peter Wong, Govind Lakkoju

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience x
Documentation Accessibility x
Related Documentation x
Conventions xi

1 Introduction to the Oracle Database Gateway for Informix

Overview of Oracle Database Gateways 1-1
About Heterogeneous Services Technology 1-1
About Oracle Database Gateway for Informix 1-2

2 Informix Gateway Features and Restrictions

Remote Insert Rowsource 2-1
Using the Pass-Through Feature 2-1
CHAR Semantics 2-2
Multi-byte Character Sets Ratio Suppression 2-3
IPv6 Support 2-3
Gateway Session IDLE Timeout 2-3
Database Compatibility Issues for Informix 2-3
- ANSI SQL Standard 2-4
- Naming Rules 2-4
  - Rules for Naming Objects 2-4
  - Object Names 2-4
  - Case Sensitivity 2-4
- Data Types 2-5
  - Binary, Byte and Text Literal Notation 2-5
  - Data Type Conversion 2-6
- Queries 2-6
  - Row Selection 2-6
  - Empty Strings 2-6
  - Empty Bind Variables 2-7

Locking 2-7
Known Restrictions 2-7
Transactional Integrity 2-8
Transaction Capability 2-8
COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors 2-9
Pass-Through Feature 2-9
Informix SMALLFLOAT and REAL Data Types 2-9
SQL Syntax 2-9
WHERE CURRENT OF Clause 2-9
CONNECT BY Clause 2-10
Use of NULL Keyword in SELECT Statement 2-10
Subqueries in INSERT Statement 2-10
Subqueries in DELETE, INSERT, and UPDATE Statements 2-10
ROWID 2-10
EXPLAIN PLAN Statement 2-10
SQL*Plus 2-10
Database Links 2-10
Gateway Data Dictionary Views 2-11
Stored Procedures 2-11
Known Problems 2-11
Encrypted Format Login 2-11
Informix BYTE and TEXT Data Types 2-11
Schema Names and PL/SQL 2-12
Data Dictionary Views and PL/SQL 2-12

3 Case Studies

Case Descriptions 3-1
Distribution Media Contents 3-1
Demonstration Files 3-2
Demonstration Requirements 3-2
Creating Demonstration Tables 3-2
Demonstration Table Definitions 3-3
Demonstration Table Contents 3-3
Case 1: Simple Queries 3-4
Case 2: A More Complex Query 3-4
Case 3: Joining Informix Tables 3-5
Case 4: Write Capabilities 3-5
DELETE Statement 3-5
UPDATE Statement 3-5
INSERT Statement 3-5

Case 5: Data Dictionary Query 3-5
Case 6: The Pass-Through Feature 3-5
UPDATE Statement 3-6
SELECT Statement 3-6

A Data Type Conversion

B Supported SQL Syntax and Functions

Supported SQL Statements B-1
DELETE B-1
INSERT B-1
SELECT B-2
UPDATE B-2
Oracle Functions B-2
Functions Not Supported by Informix B-2
Functions Supported by Informix B-2
Arithmetic Operators B-3
Comparison Operators B-3
Group Functions B-3
String Functions B-4
Pattern Matches B-4
Date Functions B-4
Other Functions B-4
Functions Supported by the Gateway B-5

C Data Dictionary

Informix System Catalog Tables C-1
Accessing the Gateway Data Dictionary C-1
Direct Queries to Informix Tables C-2
Supported Views and Tables C-2
Data Dictionary Mapping C-3
Default Column Values C-4
Gateway Data Dictionary Descriptions C-4
ALL_CATALOG C-5
ALL_COL_COMMENTS C-5
ALL_COL_PRIVS C-5
ALL_CONS_COLUMNS C-6
ALL_CONSTRAINTS C-6
ALL_IND_COLUMNS C-6

ALL_INDEXES C-7
ALL_OBJECTS C-8
ALL_SYNONYMS C-9
ALL_TAB_COLUMNS C-9
ALL_TAB_COMMENTS C-10
ALL_TAB_PRIVS C-11
ALL_TABLES C-11
ALL_USERS C-12
ALL_VIEWS C-13
COLUMN_PRIVILEGES C-13
DBA_CATALOG C-14
DBA_COL_COMMENTS C-14
DBA_OBJECTS C-14
DBA_TAB_COLUMNS C-15
DBA_TAB_COMMENTS C-16
DBA_TABLES C-16
DICT_COLUMNS C-17
DICTIONARY C-18
DUAL C-18
TABLE_PRIVILEGES C-18
USER_CATALOG C-18
USER_COL_COMMENTS C-19
USER_COL_PRIVS C-19
USER_CONS_COLUMNS C-19
USER_CONSTRAINTS C-20
USER_IND_COLUMNS C-20
USER_INDEXES C-21
USER_OBJECTS C-22
USER_SYNONYMS C-23
USER_TAB_COLUMNS C-23
USER_TAB_COMMENTS C-24
USER_TAB_PRIVS C-24
USER_TABLES C-25
USER_USERS C-26
USER_VIEWS C-26

D Initialization Parameters

Initialization Parameter File Syntax D-1
Oracle Database Gateway for Informix Initialization Parameters D-2
HS_DB_DOMAIN D-3

HS_DB_INTERNAL_NAME D-4
HS_DB_NAME D-4
HS_DESCRIBE_CACHE_HWM D-4
HS_LANGUAGE D-4
Character Sets D-5
Language D-5
Territory D-5
HS_LONG_PIECE_TRANSFER_SIZE D-6
HS_OPEN_CURSORS D-6
HS_RPC_FETCH_REBLOCKING D-6
HS_RPC_FETCH_SIZE D-7
HS_TIME_ZONE D-7
HS_TRANSACTION_MODEL D-7
IFILE D-8
HS_FDS_TIMESTAMP_MAPPING D-8
HS_FDS_DATE_MAPPING D-9
HS_FDS_CONNECT_INFO D-9
HS_FDS_RECOVERY_ACCOUNT D-9
HS_FDS_RECOVERY_PWD D-10
HS_FDS_TRACE_LEVEL D-10
HS_FDS_TRANSACTION_ISOLATION D-10
HS_FDS_TRANSACTION_LOG D-11
HS_FDS_FETCH_ROWS D-11
HS_IDLE_TIMEOUT D-11
HS_NLS_LENGTH_SEMANTICS D-11
HS_KEEP_REMOTE_COLUMN_SIZE D-12
HS_FDS_REMOTE_DB_CHARSET D-12
HS_FDS_SUPPORT_STATISTICS D-13
HS_FDS_SQLLEN_INTERPRETATION D-13
HS_FDS_ARRAY_EXEC D-13

Index

List of Tables

A-1	Data Type Mapping and Restrictions	A-1
C-1	Oracle Data Dictionary View Names and Informix Equivalents	C-3
C-2	ALL_CATALOG	C-5
C-3	ALL_COL_COMMENTS	C-5
C-4	ALL_COL_PRIVS	C-5
C-5	ALL_CONS_COLUMNS	C-6
C-6	ALL_CONSTRAINTS	C-6
C-7	ALL_IND_COLUMNS	C-6
C-8	ALL_INDEXES	C-7
C-9	ALL_OBJECTS	C-8
C-10	ALL_SYNONYMS	C-9
C-11	ALL_TAB_COLUMNS	C-9
C-12	ALL_TAB_COMMENTS	C-10
C-13	ALL_TAB_PRIVS	C-11
C-14	ALL_TABLES	C-11
C-15	ALL_USERS	C-12
C-16	ALL_VIEWS	C-13
C-17	COLUMN_PRIVILEGES	C-13
C-18	DBA_CATALOG	C-14
C-19	DBA_COL_COMMENTS	C-14
C-20	DBA_OBJECTS	C-14
C-21	DBA_TAB_COLUMNS	C-15
C-22	DBA_TAB_COMMENTS	C-16
C-23	DBA_TABLES	C-16
C-24	DICT_COLUMNS	C-17
C-25	DICTIONARY	C-18
C-26	DUAL	C-18
C-27	TABLE_PRIVILEGES	C-18
C-28	USER_CATALOG	C-18
C-29	USER_COL_COMMENTS	C-19
C-30	USER_COL_PRIVS	C-19
C-31	USER_CONS_COLUMNS	C-19
C-32	USER_CONSTRAINTS	C-20
C-33	USER_IND_COLUMNS	C-20
C-34	USER_INDEXES	C-21

C-35 USER_OBJECTS C-22
C-36 USER_SYNONYMS C-23
C-37 USER_TAB_COLUMNS C-23
C-38 USER_TAB_COMMENTS C-24
C-39 USER_TAB_PRIVS C-24
C-40 USER_TABLES C-25
C-41 USER_USERS C-26
C-42 USER_VIEWS C-26

Preface

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

This preface covers the following topics:

- Audience
- Documentation Accessibility
- Related Documentation
- Conventions

Audience

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

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documentation

For more information, see the following documents:

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

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

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to the Oracle Database Gateway for Informix

Oracle Database Gateways provide the ability to transparently access data residing in a non-Oracle system from an Oracle environment. The following topics briefly cover Heterogeneous Services, the technology that the Oracle Database Gateway for Informix is based on.

> **See Also:**
>
> *Oracle Database Heterogeneous Connectivity User's Guide* to get a good understanding of generic gateway technology, Heterogeneous Services, and how Oracle Database Gateways fit in the picture.

Overview of Oracle Database Gateways

Oracle Database Gateways provide the ability to transparently access data residing in a non-Oracle system from an Oracle environment.

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 Database 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 Oracle Database Gateway agent, enables transparent access to non-Oracle systems from an Oracle environment.

About 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 that 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 User's Guide for more information about Heterogeneous Services.

About Oracle Database Gateway for Informix

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 can be installed on any machine independent of the Oracle or non-Oracle database. It 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

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.

Remote Insert Rowsource

A remote insert rowsource feature allows remote insert requiring local Oracle data to work through the Oracle database and Oracle Database Gateway. This functionality requires that the Oracle database and the Oracle Database Gateway to be version 12.2 or later.

By Oracle Database design, some distributed statement must be executed at the database link site. But in certain circumstances, there is data needed to execute these queries that must be fetched from the originating Oracle Database. Under homogeneous connections, the remote Oracle database would call back the source Oracle database for such data. But in heterogeneous connections, this is not viable, because this means that the Foreign Data Store would have to query call back functions, or data, that can only be provided by the Oracle instance that issued the query. In general, these kinds of statements are not something that can be supported through the Oracle Database Gateway.

The following categories of SQL statements results in a callback:

- Any DML with a sub-select, which refers to a table in Oracle database.
- Any `DELETE`, `INSERT`, `UPDATE` or "`SELECT... FOR UPDATE...`" SQL statement containing SQL functions or statements that needs to be executed at the originating Oracle database.

  These SQL functions include `USER`, `USERENV`, and `SYSDATE`; and involve the selection of data from the originating Oracle database.
- Any SQL statement that involves a table in Oracle database, and a `LONG` or `LOB` column in a remote table.

An example of a remote `INSERT` statement that can work through the remote insert rowsource feature is as follows:

```
INSERT INTO gateway_table@gateway_link select * from local_table;
```

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.

> **Note:**
>
> It is recommended that you `COMMIT` after each DDL statement in the pass-through.

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

CHAR Semantics

This feature allows the gateway to optionally run in `CHAR` semantics mode.

Rather than always describing Informix `CHAR` columns as `CHAR(n BYTE)`, this feature describes them as `CHAR(n CHAR)` and `VARCHAR(n CHAR)`. The concept is similar to Oracle database `CHAR` semantics. You need to specify `HS_NLS_LENGTH_SEMANTICS=CHAR` gateway parameter to activate this option.

> **See Also:**
>
> Initialization Parameters

Multi-byte Character Sets Ratio Suppression

This feature optionally suppresses the ratio expansion from Informix database to Oracle database involving multi-byte character set.

By default, Oracle gateways assume the worst ratio to prevent data being truncated or insufficient buffer size situation. However, if you have specific knowledge of your Informix database and do not want the expansion to occur, you can specify `HS_KEEP_REMOTE_COLUMN_SIZE` parameter to suppress the expansion.

> **See Also:**
>
> Initialization Parameters

IPv6 Support

Besides full IPv6 support between Oracle databases and the gateway, IPv6 is also supported between this gateway and Informix database.

> **See Also:**
>
> Initialization Parameters for more detail about the `HS_FDS_CONNECT_INFO` parameter.

Gateway Session IDLE Timeout

You can optionally choose to terminate long idle gateway sessions automatically with the gateway parameter `HS_IDLE_TIMEOUT`. Specifically, when a gateway session is idle for more than the specified time limit, the gateway session is terminated with any pending update rolled back. Refer to the `HS_IDLE_TIMEOUT` parameter in Initialization Parameters for more detail.

Database Compatibility Issues for Informix

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

- 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 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 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 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.

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 ('0xff');
```

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:

```
SQL> 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:

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

> **See Also:**
>
> 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 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 the following possible scenarios can affect Oracle applications that access Informix through the gateway:

- 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", contact Oracle Support Services. The following topics describe 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
- 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 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.

If your Informix system does not use logging, then you cannot use the default transaction capability. It is strongly recommended that you enable logging in your Informix system. If you are not using logging, and have read only requirement, then set `HS_TRANSACTION_MODEL=READ_ONLY_AUTOCOMMIT` in the gateway initialization parameter file. If you are not using logging, and you require to update the Informix database, then set `HS_TRANSACTION_MODEL=SINGLE_SITE_AUTOCOMMIT` in the gateway initialization parameter file.

> **See Also:**
>
> Initialization Parameters and the *Oracle Database Heterogeneous Connectivity User'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`.

SQL Syntax

Restrictions on SQL syntax are listed as follows:

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

The known problems are as follows:

- 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 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 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.
- Data less than 32,739 bytes cannot be inserted into `BYTE` and `TEXT` columns using bind variables.

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

> **See Also:**
>
> Supported SQL Syntax and Functions.

Schema Names and PL/SQL

If you do not prefix an 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.

3

Case Studies

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

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

Topics:

- Case Descriptions
- Distribution Media 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)

Distribution Media Contents

The distribution media 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/dg4ifmx/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 database has an account named SCOTT with a password of TIGER
- The Oracle database 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 $ORACLE_HOME/dg4ifmx/demo$ 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.

A

Data Type Conversion

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

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.

Table A-1 Data Type Mapping and Restrictions

Informix	Oracle	Criteria	If Oracle uses large varchar (32k)
BLOB	LONG RAW	-	
BOOLEAN	NUMBER(3)		
BYTE	LONG RAW	-	
CLOB	LONG	-	
CHAR	CHAR	-	
DATE	DATE	-	
DATETIME YEAR TO DAY	DATE		
DATETIME YEAR TO FRACTION	DATE		
DATETIME YEAR TO SECOND	DATE		
DATETIME HOUR TO SECOND	CHAR(15)		
DATETIME HOUR TO FRACTION	CHAR(15)		
DECIMAL	NUMBER(p[,s])	-	
FLOAT	FLOAT(53)	-	
INT8	NUMBER(19)		
INTEGER	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647	
INTERVAL YEAR() TO YEAR	INTERVAL YEAR TO MONTH		
INTERVAL MONTH() TO MONTH	INTERVAL YEAR TO MONTH		

Table A-1 (Cont.) Data Type Mapping and Restrictions

Informix	Oracle	Criteria	If Oracle uses large varchar (32k)
INTERVAL YEAR() TO MONTH	INTERVAL YEAR TO MONTH		
INTERVAL DAY() TO DAY	INTERVAL DAY TO SECOND		
INTERVAL HOUR() TO HOUR	INTERVAL DAY TO SECOND		
INTERVAL MINUTE() TO MINUTE	INTERVAL DAY TO SECOND		
INTERVAL SECOND() TO SECOND	INTERVAL DAY TO SECOND		
INTERVAL SECOND() TO FRACTION	INTERVAL DAY TO SECOND		
INTERVAL FRACTION TO FRACTION	INTERVAL DAY TO SECOND		
INTERVAL DAY() TO HOUR	INTERVAL DAY TO SECOND		
INTERVAL DAY() TO MINUTE	INTERVAL DAY TO SECOND		
INTERVAL DAY() TO SECOND	INTERVAL DAY TO SECOND		
INTERVAL DAY() TO FRACTION	INTERVAL DAY TO SECOND		
INTERVAL HOUR() TO MINUTE	INTERVAL DAY TO SECOND		
INTERVAL HOUR() TO SECOND	INTERVAL DAY TO SECOND		
INTERVAL HOUR() TO FRACTION	INTERVAL DAY TO SECOND		
INTERVAL MINUTE() TO SECOND	INTERVAL DAY TO SECOND		
INTERVAL MINUTE() TO FRACTION	INTERVAL DAY TO SECOND		
LVARCHAR	VARCHAR2	N < = 4000	N <= 32767
MONEY	NUMBER(p[,s])	-	
NCHAR	CHAR	-	
NVARCHAR	VARCHAR2	N < = 4000	N <= 32767
SERIAL	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647	
SERIAL8	NUMBER(19)	-	

Table A-1 (Cont.) Data Type Mapping and Restrictions

Informix	Oracle	Criteria	If Oracle uses large varchar (32k)
SMALLFLOAT	FLOAT(24)	Precision is 6	
SMALLINT	NUMBER(5)		
TEXT	LONG	-	
VARCHAR	VARCHAR2	N < = 4000 If a length is not specified as part of VARCHAR, the data type is converted to VARCHAR2(1)	N <= 32767

B
Supported SQL Syntax and Functions

The following topics describe supported SQL Syntax and Functions:

- 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 Language Reference* for 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" 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" 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" 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 if the statement is a `SELECT` statement. That is, the Oracle database 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 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

Oracle	Informix
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

> **Note:**
>
> 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 Database Gateway for Informix translates a query that refers to an Oracle database 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. The gateway data dictionary is similar to the Oracle database data dictionary in appearance and use.

Topics:

- Data Dictionary Mapping
- Gateway Data Dictionary Descriptions

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 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 data dictionary table.

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

> **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
USER_SYNONYMS	USER_TAB_COLUMNS
USER_TAB_COMMENTS	USER_TAB_PRIVS
USER_TABLES	USER_USERS

Supported Views and Tables	Supported Views and Tables
USER_VIEWS	

No other Oracle database data dictionary tables or views are supported. If you use a view not on the list, you receive the Oracle database 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 data dictionary. The Oracle database 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 data dictionary tables and views. Any default value is shown to the right of an item, but this is not information returned by `DESCRIBE`.

ALL_CATALOG

Table C-2 ALL_CATALOG

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

ALL_COL_COMMENTS

Table C-3 ALL_COL_COMMENTS

Name	Type	Value
OWNER	VARCHAR2(32)	-
TABLE_NAME	VARCHAR2(128)	-
COLUMN_NAME	VARCHAR2(128)	-
COMMENTS	CHAR(1)	" "

ALL_COL_PRIVS

Table C-4 ALL_COL_PRIVS

Name	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"

ALL_CONS_COLUMNS

Table C-5 ALL_CONS_COLUMNS

Name	Type	Value
OWNER	VARCHAR2(32)	-
CONSTRAINT_NAME	VARCHAR2(128)	-
TABLE_NAME	VARCHAR2(128)	-
COLUMN_NAME	VARCHAR2(128)	-
POSITION	NUMBER(10)	0

ALL_CONSTRAINTS

Table C-6 ALL_CONSTRAINTS

Name	Value
OWNER	-
CONSTRAINT_NAME	-
CONSTRAINT_TYPE	"R" or "P" or "U" or "C"
TABLE_NAME	-
SEARCH_CONDITION	""
R_OWNER	""
R_CONSTRAINT_NAME	""
DELETE_RULE	""
STATUS	""
DEFERRABLE	""
DEFERRED	""
VALIDATED	""
GENERATED	""
BAD	""
RELY	""
LAST_CHANGE	-

ALL_IND_COLUMNS

Table C-7 ALL_IND_COLUMNS

Name	Type	Value
INDEX_OWNER	VARCHAR2(32)	-

Table C-7 (Cont.) ALL_IND_COLUMNS

Name	Type	Value
INDEX_NAME	VARCHAR2(128)	-
TABLE_OWNER	VARCHAR2(32)	-
TABLE_NAME	VARCHAR2(128)	-
COLUMN_NAME	VARCHAR2(128)	-
COLUMN_POSITION	NUMBER(10)	0
COLUMN_LENGTH	NUMBER(10)	0
DESCEND	CHAR(1)	" "

ALL_INDEXES

Table C-8 ALL_INDEXES

Name	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
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

Table C-8 (Cont.) ALL_INDEXES

Name	Type	Value
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
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

ALL_OBJECTS

Table C-9 ALL_OBJECTS

Name	Type	Value
OWNER	VARCHAR2(32)	-

Table C-9 (Cont.) ALL_OBJECTS

Name	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
SECONDARY	VARCHAR2(1)	NULL

ALL_SYNONYMS

Table C-10 ALL_SYNONYMS

Name	Type	Value
OWNER	VARCHAR2(32)	-
SYNONYM_NAME	VARCHAR2(128)	-
TABLE_OWNER	VARCHAR2(32)	-
TABLE_NAME	VARCHAR2(128)	-
DB_LINK	CHAR(1)	NULL

ALL_TAB_COLUMNS

Table C-11 ALL_TAB_COLUMNS

Name	Type	Value
OWNER	VARCHAR2(32)	-
TABLE_NAME	VARCHAR2(128)	-
COLUMN_NAME	VARCHAR2(128)	-
DATA_TYPE	VARCHAR2(8)	-
DATA_TYPE_MOD	VARCHAR2(1)	NULL

Table C-11 (Cont.) ALL_TAB_COLUMNS

Name	Type	Value
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

ALL_TAB_COMMENTS

Table C-12 ALL_TAB_COMMENTS

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

ALL_TAB_PRIVS

Table C-13 ALL_TAB_PRIVS

Name	Type	Value
GRANTOR	VARCHAR2(32)	-
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"

ALL_TABLES

Table C-14 ALL_TABLES

Name	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

Table C-14 (Cont.) ALL_TABLES

Name	Type	Value
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
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

ALL_USERS

Table C-15 ALL_USERS

Name	Type	Value
USERNAME	VARCHAR2(32)	-
USER_ID	NUMBER	0

Table C-15 (Cont.) ALL_USERS

Name	Type	Value
CREATED	DATE	SYSDATE

ALL_VIEWS

Table C-16 ALL_VIEWS

Name	Type	Value
OWNER	VARCHAR2(32)	-
VIEW_NAME	VARCHAR2(128)	-
TEXT_LENGTH	NUMBER(10)	64
TEXT	VARCHAR2(64)	-
TYPE_TEXT_LENGTH	NUMBER(10)	0
TYPE_TEXT	CHAR(1)	""
OID_TEXT_LENGTH	NUMBER(10)	0
OID_TEXT	CHAR(1)	""
VIEW_TYPE_OWNER	CHAR(1)	""
VIEW_TYPE	CHAR(1)	""

COLUMN_PRIVILEGES

Table C-17 COLUMN_PRIVILEGES

Name	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

DBA_CATALOG

Table C-18 DBA_CATALOG

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

DBA_COL_COMMENTS

Table C-19 DBA_COL_COMMENTS

Name	Type	Value
OWNER	VARCHAR2(32)	-
TABLE_NAME	VARCHAR2(128)	-
COLUMN_NAME	VARCHAR2(128)	-
COMMENTS	CHAR(1)	" "

DBA_OBJECTS

Table C-20 DBA_OBJECTS

Name	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

Table C-20 (Cont.) DBA_OBJECTS

Name	Type	Value
SECONDARY	VARCHAR2(1)	NULL

DBA_TAB_COLUMNS

Table C-21 DBA_TAB_COLUMNS

Name	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
CHAR_COL_DECL_LENGTH	NUMBER	0
GLOBAL_STATS	VARCHAR2(1)	NULL
USER_STATS	VARCHAR2(1)	NULL
AVG_COL_LEN	NUMBER	0

DBA_TAB_COMMENTS

Table C-22 DBA_TAB_COMMENTS

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

DBA_TABLES

Table C-23 DBA_TABLES

Name	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

Table C-23 (Cont.) DBA_TABLES

Name	Type	Value
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

DICT_COLUMNS

Table C-24 DICT_COLUMNS

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

DICTIONARY

Table C-25 DICTIONARY

Name	Type	Value
TABLE_NAME	VARCHAR2(128)	-
COMMENTS	CHAR(1)	" "

DUAL

Table C-26 DUAL

Name	Type	Value
DUMMY	VARCHAR2(1)	"X"

TABLE_PRIVILEGES

Table C-27 TABLE_PRIVILEGES

Name	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

USER_CATALOG

Table C-28 USER_CATALOG

Name	Type	Value
TABLE_NAME	VARCHAR2(128)	-

Table C-28 (Cont.) USER_CATALOG

Name	Type	Value
TABLE_TYPE	VARCHAR2(7)	"TABLE" or "VIEW" or "SYNONYM"

USER_COL_COMMENTS

Table C-29 USER_COL_COMMENTS

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

USER_COL_PRIVS

Table C-30 USER_COL_PRIVS

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

USER_CONS_COLUMNS

Table C-31 USER_CONS_COLUMNS

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

USER_CONSTRAINTS

Table C-32 USER_CONSTRAINTS

Name	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	

USER_IND_COLUMNS

Table C-33 USER_IND_COLUMNS

Name	Type	Value
INDEX_NAME	VARCHAR2(128)	-
TABLE_NAME	VARCHAR2(128)	-
COLUMN_NAME	VARCHAR2(128)	-
COLUMN_POSITION	NUMBER	0
COLUMN_LENGTH	NUMBER	0
DESCEND	VARCHAR2(1)	-

USER_INDEXES

Table C-34 USER_INDEXES

Name	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
SAMPLE_SIZE	NUMBER	0

Table C-34 (Cont.) USER_INDEXES

Name	Type	Value
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

USER_OBJECTS

Table C-35 USER_OBJECTS

Name	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

Table C-35 (Cont.) USER_OBJECTS

Name	Type	Value
STATUS	VARCHAR2(1)	NULL
TEMPORARY	VARCHAR2(1)	NULL
GENERATED	VARCHAR2(1)	NULL
SECONDARY	VARCHAR2(1)	NULL

USER_SYNONYMS

Table C-36 USER_SYNONYMS

Name	Type	Value
SYNONYM_NAME	VARCHAR2(128)	-
TABLE_OWNER‘	VARCHAR2(32)	-
TABLE_NAME	VARCHAR2(128)	-
DB_LINK	VARCHAR2(1)	NULL

USER_TAB_COLUMNS

Table C-37 USER_TAB_COLUMNS

Name	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

Table C-37 (Cont.) USER_TAB_COLUMNS

Name	Type	Value
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

USER_TAB_COMMENTS

Table C-38 USER_TAB_COMMENTS

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

USER_TAB_PRIVS

Table C-39 USER_TAB_PRIVS

Name	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"

USER_TABLES

Table C-40 USER_TABLES

Name	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
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')

Table C-40 (Cont.) USER_TABLES

Name	Type	Value
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

USER_USERS

Table C-41 USER_USERS

Name	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
EXTERNAL_NAME	VARCHAR2(1)	NULL

USER_VIEWS

Table C-42 USER_VIEWS

Name	Type	Value
VIEW_NAME	VARCHAR2(128)	-
TEXT_LENGTH	NUMBER	64

Table C-42 (Cont.) USER_VIEWS

Name	Type	Value
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
Initialization Parameters

The Oracle database initialization parameters in the `init.ora` file are distinct from gateway initialization parameters. Set the gateway parameters in the initialization parameter file using an agent-specific mechanism, or set them in the Oracle data dictionary using the `DBMS_HS` package. The gateway initialization parameter file must be available when the gateway is started.

The following topics contain a list of the gateway initialization parameters that can be set for each gateway and their description. The topics also describe the initialization parameter file syntax.

- Initialization Parameter File Syntax
- Oracle Database Gateway for Informix Initialization Parameters

Initialization Parameter File Syntax

The syntax for the initialization parameter file is as follows:

- The file is a sequence of commands.
- Each command should start on a separate line.
- End of line is considered a command terminator (unless escaped with a backslash).
- If there is a syntax error in an initialization parameter file, none of the settings take effect.
- Set the parameter values as follows:

  `[SET][PRIVATE] parameter=value`

  Where:

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

  value is the initialization parameter value. It is case-sensitive. An initialization parameter value is either:

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

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

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

* \\ inserts a backslash

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

For example, to enable tracing for an agent, set the HS_FDS_TRACE_LEVEL initialization parameter as follows:

```
HS_FDS_TRACE_LEVEL=ON
```

SET and PRIVATE are optional keywords. You cannot use either as an initialization parameter name. Most parameters are needed only as initialization parameters, so you usually do not need to use the SET or PRIVATE keywords. If you do not specify either SET or PRIVATE, the parameter is used only as an initialization parameter for the agent.

SET specifies that, in addition to being used as an initialization parameter, the parameter value is set as an environment variable for the agent process. Use SET for parameter values that the drivers or non-Oracle system need as environment variables.

PRIVATE specifies that the initialization parameter should be private to the agent and should not be uploaded to the Oracle database. Most initialization parameters should not be private. If, however, you are storing sensitive information like a password in the initialization parameter file, then you may not want it uploaded to the server because the initialization parameters and values are not encrypted when uploaded. Making the initialization parameters private prevents the upload from happening and they do not appear in dynamic performance views. Use PRIVATE for the initialization parameters only if the parameter value includes sensitive information such as a user name or password.

SET PRIVATE specifies that the parameter value is set as an environment variable for the agent process and is also private (not transferred to the Oracle database, not appearing in dynamic performance views or graphical user interfaces).

Oracle Database Gateway for Informix Initialization Parameters

The initialization file parameters that can be set for the Oracle Database Gateway for Informix are as follows:

- HS_DB_DOMAIN
- HS_DB_INTERNAL_NAME
- HS_DB_NAME
- HS_DESCRIBE_CACHE_HWM
- HS_LANGUAGE
- HS_LONG_PIECE_TRANSFER_SIZE
- HS_OPEN_CURSORS
- HS_RPC_FETCH_REBLOCKING
- HS_RPC_FETCH_SIZE
- HS_TIME_ZONE

- HS_FDS_TRANSACTION_ISOLATION
- HS_TRANSACTION_MODEL
- IFILE
- HS_FDS_TIMESTAMP_MAPPING
- HS_FDS_DATE_MAPPING
- HS_FDS_CONNECT_INFO
- HS_FDS_RECOVERY_ACCOUNT
- HS_FDS_RECOVERY_PWD
- HS_FDS_TRACE_LEVEL
- HS_FDS_TRANSACTION_LOG
- HS_FDS_FETCH_ROWS
- HS_IDLE_TIMEOUT
- HS_NLS_LENGTH_SEMANTICS
- HS_KEEP_REMOTE_COLUMN_SIZE
- HS_FDS_REMOTE_DB_CHARSET
- HS_FDS_SUPPORT_STATISTICS
- HS_FDS_SQLLEN_INTERPRETATION
- HS_FDS_ARRAY_EXEC

HS_DB_DOMAIN

Property	Description
Default value	`WORLD`
Range of values	1 to 199 characters

Specifies a unique network sub-address for a non-Oracle system. The `HS_DB_DOMAIN` initialization parameter is similar to the `DB_DOMAIN` initialization parameter, described in the *Oracle Database Reference*. The `HS_DB_DOMAIN` initialization parameter is required if you use the Oracle Names server. The `HS_DB_NAME` and `HS_DB_DOMAIN` initialization parameters define the global name of the non-Oracle system.

> **Note:**
>
> The `HS_DB_NAME` and `HS_DB_DOMAIN` initialization parameters must combine to form a unique address in a cooperative server environment.

HS_DB_INTERNAL_NAME

Property	Description
Default value	`01010101`
Range of values	1 to 16 hexadecimal characters

Specifies a unique hexadecimal number identifying the instance to which the Heterogeneous Services agent is connected. This parameter's value is used as part of a transaction ID when global name services are activated. Specifying a nonunique number can cause problems when two-phase commit recovery actions are necessary for a transaction.

HS_DB_NAME

Property	Description
Default value	`HO`
Range of values	1 to 8 characters

Specifies a unique alphanumeric name for the data store given to the non-Oracle system. This name identifies the non-Oracle system within the cooperative server environment. The `HS_DB_NAME` and `HS_DB_DOMAIN` initialization parameters define the global name of the non-Oracle system.

HS_DESCRIBE_CACHE_HWM

Property	Description
Default value	`100`
Range of values	1 to `4000`

Specifies the maximum number of entries in the describe cache used by Heterogeneous Services. This limit is known as the describe cache high water mark. The cache contains descriptions of the mapped tables that Heterogeneous Services reuses so that it does not have to re-access the non-Oracle data store.

If you are accessing many mapped tables, increase the high water mark to improve performance. Increasing the high water mark improves performance at the cost of memory usage.

HS_LANGUAGE

Property	Description
Default value	System-specific
Range of values	Any valid language name (up to 255 characters)

Provides Heterogeneous Services with character set, language, and territory information of the non-Oracle data source. The value must use the following format:

language[*_territory.character_set*]

> **Note:**
>
> The globalization support initialization parameters affect error messages, the data for the SQL Service, and parameters in distributed external procedures.

Character Sets

Ideally, the character sets of the Oracle database and the non-Oracle data source are the same. In almost all cases, HS_LANGUAGE should be set exactly the same as Oracle database character set for optimal character set mapping and performance. If they are not the same, Heterogeneous Services attempts to translate the character set of the non-Oracle data source to the Oracle database character set, and back again. The translation can degrade performance. In some cases, Heterogeneous Services cannot translate a character from one character set to another.

> **Note:**
>
> The specified character set must be a superset of the operating system character set on the platform where the agent is installed.

As more Oracle databases and non-Oracle databases use Unicode as database character sets, it is preferable to also run the gateway in Unicode character set. To do so, you must set HS_LANGUAGE=AL32UTF8. However, when the gateway runs on Windows, the Microsoft ODBC Driver Manager interface can exchange data only in the double-byte character set, UCS2. This results in extra ratio expansion of described buffer and column sizes. Refer to HS_FDS_REMOTE_DB_CHARSET for instruction on how to adjust to correct sizes.

Language

The language component of the HS_LANGUAGE initialization parameter determines:

- Day and month names of dates
- AD, BC, PM, and AM symbols for date and time
- Default sorting mechanism

Note that Oracle does not determine the language for error messages for the generic Heterogeneous Services messages (ORA-25000 through ORA-28000). These are controlled by the session settings in the Oracle database.

Territory

The territory clause specifies the conventions for day and week numbering, default date format, decimal character and group separator, and ISO and local currency

symbols. Note that the level of globalization support between the Oracle database and the non-Oracle data source depends on how the gateway is implemented.

HS_LONG_PIECE_TRANSFER_SIZE

Property	Description
Default value	64 KB
Range of values	Any value up to 2 GB

Sets the size of the piece of `LONG` data being transferred. A smaller piece size means less memory requirement, but more round-trips to fetch all the data. A larger piece size means fewer round-trips, but more of a memory requirement to store the intermediate pieces internally. Thus, the initialization parameter can be used to tune a system for the best performance, with the best trade-off between round-trips and memory requirements, and network latency or response time.

HS_OPEN_CURSORS

Property	Description
Default value	`50`
Range of values	1 to the value of Oracle's `OPEN_CURSORS` initialization parameter

Defines the maximum number of cursors that can be open on one connection to a non-Oracle system instance.

The value never exceeds the number of open cursors in the Oracle database. Therefore, setting the same value as the `OPEN_CURSORS` initialization parameter in the Oracle database is recommended.

HS_RPC_FETCH_REBLOCKING

Property	Description
Default value	`ON`
Range of values	`OFF` or `ON`

Controls whether Heterogeneous Services attempts to optimize performance of data transfer between the Oracle database and the Heterogeneous Services agent connected to the non-Oracle data store.

The following values are possible:

- `OFF` disables reblocking of fetched data so that data is immediately sent from agent to server.
- `ON` enables reblocking, which means that data fetched from the non-Oracle system is buffered in the agent and is not sent to the Oracle database until the amount of fetched data is equal or higher than the value of `HS_RPC_FETCH_SIZE` initialization parameter. However, any buffered data is returned immediately when a fetch indicates that no more data exists or when the non-Oracle system reports an error.

HS_RPC_FETCH_SIZE

Property	Description
Default value	50000
Range of values	1 to 10000000

Tunes internal data buffering to optimize the data transfer rate between the server and the agent process.

Increasing the value can reduce the number of network round-trips needed to transfer a given amount of data, but also tends to increase data bandwidth and to reduce latency as measured between issuing a query and completion of all fetches for the query. Nevertheless, increasing the fetch size can increase latency for the initial fetch results of a query, because the first fetch results are not transmitted until additional data is available.

HS_TIME_ZONE

Property	Description
Default value for '[+|-]hh:mm'	Derived from the NLS_TERRITORY initialization parameter
Range of values for '[+|-]hh:mm'	Any valid datetime format mask

Specifies the default local time zone displacement for the current SQL session. The format mask, [+|-]hh:mm, is specified to indicate the hours and minutes before or after UTC (Coordinated Universal Time—formerly Greenwich Mean Time). For example:

```
HS_TIME_ZONE = [+ | -] hh:mm
```

HS_TRANSACTION_MODEL

Property	Description
Default Value	COMMIT_CONFIRM
Range of Values	COMMIT_CONFIRM, READ_ONLY, SINGLE_SITE, READ_ONLY_AUTOCOMMIT, SINGLE_SITE_AUTOCOMMIT

Specifies the type of transaction model that is used when the non-Oracle database is updated by a transaction.

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the non-Oracle database and allows the gateway to be part of a distributed update. To use the commit-confirm model, the following items must be created in the non-Oracle database:
  - Transaction log table. The default table name is HS_TRANSACTION_LOG. A different name can be set using the HS_FDS_TRANSACTION_LOG parameter. The

transaction log table must be granted `SELECT`, `DELETE`, and `INSERT` privileges set to public.

  - Recovery account. The account name is assigned with the `HS_FDS_RECOVERY_ACCOUNT` parameter.
  - Recovery account password. The password is assigned with the `HS_FDS_RECOVERY_PWD` parameter.

- `READ_ONLY` provides read access to the non-Oracle database.
- `SINGLE_SITE` provides read and write access to the non-Oracle database. However, the gateway cannot participate in distributed updates.
- `READ_ONLY_AUTOCOMMIT` provides read only access to the non-Oracle database that does not use logging.
- `SINGLE_SITE_AUTOCOMMIT` provides read and write access to the non-Oracle database without logging. The gateway cannot participate in distributed updates. Moreover, any update to the non-Oracle database is committed immediately.

IFILE

Property	Description
Default value	None
Range of values	Valid parameter file names

Use the `IFILE` initialization parameter to embed another initialization file within the current initialization file. The value should be an absolute path and should not contain environment variables. The three levels of nesting limit do not apply.

> **See Also:**
>
> *Oracle Database Reference*

HS_FDS_TIMESTAMP_MAPPING

Property	Description
Default Value	`DATE`
Range of Values	`CHAR|DATE|TIMESTAMP`
Syntax	`HS_FDS_TIMESTAMP_MAPPING={`*`CHAR`*`|`*`DATE`*`|`*`TIMESTAMP`*`}`

If set to `CHAR` , then non-Oracle target timestamp would be mapped to `CHAR(26)`. If set to `DATE` (default), then non-Oracle target timestamp would be mapped to Oracle `DATE`. If set to `TIMESTAMP`, then non-Oracle target timestamp would be mapped to Oracle `TIMESTAMP`.

HS_FDS_DATE_MAPPING

Property	Description
Default Value	DATE
Range of Values	DATE|CHAR
Syntax	HS_FDS_DATE_MAPPING={*DATE*|*CHAR*}

If set to CHAR, then non-oracle target date would be mapped to CHAR(10). If set to DATE, then non-Oracle target date would be mapped to Oracle date.

HS_FDS_CONNECT_INFO

Property	Description
Default Value	None
Range of Values	Not applicable

HS_FDS_CONNECT_INFO that describes the connection to the non-Oracle system.

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

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

where, *host_name* is the host name or IP address of the machine hosting the Informix database, *port_number* is the port number of the Informix database server, *server_name* is the name of the Informix database server, and *database_name* is the Informix database name.

This release supports IPv6 format, so you can enter IPv6 format in place of hostname, but you need to wrap square brackets around the IPv6 specification.

For example,

HS_FDS_CONNECT_INFO=[2001:0db8:20c:f1ff:fec6:38af]:port_number/…

HS_FDS_RECOVERY_ACCOUNT

Property	Description
Default Value	RECOVER
Range of values	Any valid user ID

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

The name of the recovery account is case-sensitive.

HS_FDS_RECOVERY_PWD

Property	Description
Default Value	RECOVER
Range of values	Any valid password

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

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

HS_FDS_TRACE_LEVEL

Property	Description
Default Value	OFF
Range of values	OFF, ON, DEBUG

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

The following values are valid:

- OFF disables the tracing of error messages.
- ON enables the tracing of error messages that occur when you encounter problems. The results are written by default to a gateway log file in LOG directory where the gateway is installed.
- DEBUG enables the tracing of detailed error messages that can be used for debugging.

HS_FDS_TRANSACTION_ISOLATION

Property	Description
Default Value	READ_COMMITTED
Range of Values	{READ_UNCOMMITTED|READ_COMMITTED|REPEATABLE_READ|SERIALIZABLE}
Syntax	HS_FDS_ISOLATION_LEVEL={{READ_UNCOMMITTED|READ_COMMITTED|REPEATABLE_READ|SERIALIZABLE}

HS_FDS_TRANSACTION_ISOLATION specifies the isolation level that is used for the transaction that the gateway opens on the non-Oracle database.

The isolation levels of READ_UNCOMMITTED, READ_COMMITTED, REPEATABLE_READ, and SERIALIZABLE are the four isolation levels defined in the SQL standard and adopted by both ANSI and ISO/IEC. For additional information regarding them, see *Oracle Database Concepts*.

Use caution when specifying an isolation level lower than the Oracle transaction isolation level being used, as the gateway transaction will have different Preventable Read Phenomena from what will occur in the Oracle database transaction.

HS_FDS_TRANSACTION_LOG

Property	Description
Default Value	`HS_TRANSACTION_LOG`
Range of Values	Any valid table name

Specifies the name of the table created in the non-Oracle system for logging transactions. For more information about the transaction model, see the `HS_TRANSACTION_MODEL` parameter.

HS_FDS_FETCH_ROWS

Property	Description
Default Value	`100`
Range of Values	Any integer between 1 and `1000`
Syntax	`HS_FDS_FETCH_ROWS=`*`num`*

`HS_FDS_FETCH_ROWS` specifies the fetch array size. This is the number of rows to be fetched from the non-Oracle database and to return to Oracle database at one time. This parameter will be affected by the `HS_RPC_FETCH_SIZE` and `HS_RPC_FETCH_REBLOCKING` parameters.

HS_IDLE_TIMEOUT

Property	Description
Default Value	0 (no timeout)
Range of Values	0-9999 (minutes)
Syntax	`HS_IDLE_TIMEOUT=`*`num`*

This feature is only available for Oracle Net TCP protocol. When there is no activity for a connected gateway session for this specified time period, the gateway session would be terminated automatically with pending update (if any) rolled back.

HS_NLS_LENGTH_SEMANTICS

Property	Description
Default Value	`BYTE`
Range of Values	`BYTE | CHAR`
Syntax	`HS_NLS_LENGTH_SEMANTICS = { BYTE | CHAR }`

This release of gateway has Character Semantics functionality equivalent to the Oracle Database Character Semantics, that is, NLS_LENGTH_SEMANTICS. When HS_NLS_LENGTH_SEMANTICS is set to CHAR, the (VAR)CHAR columns of Informix database are to be interpreted as having CHAR semantics. The only situation the gateway does not honor the HS_NLS_LENGTH_SEMANTICS=CHAR setting is when both Oracle database and the gateway are on the same multi-byte character set.

HS_KEEP_REMOTE_COLUMN_SIZE

Property	Description
Default Value	OFF
Range of Values	OFF | LOCAL |REMOTE | ALL
Syntax	HS_KEEP_REMOTE_COLUMN_SIZE = OFF | LOCAL |REMOTE | ALL
Parameter type	String

HS_KEEP_REMOTE_COLUMN_SIZE specifies whether to suppress ratio expansion when computing the length of (VAR)CHAR datatypes during data conversion from non-Oracle database to the gateway, and then to the Oracle database. When it is set to REMOTE, the expansion is suppressed between the non-Oracle database and the gateway. When it is set to LOCAL, the expansion is suppressed between the gateway and the Oracle database. When it is set to ALL, the expansion is suppressed from the non-Oracle database to the Oracle database.

When the parameter is set, the expansion is suppressed when reporting the remote column size, calculating the implicit resulting buffer size, and instantiating in the local Oracle database. This has effect only for remote column size from non-Oracle database to Oracle database. If the gateway runs on Windows and HS_LANGUAGE=AL32UTF8, then you must not specify this parameter, as it would influence other ratio related parameter operation. It has no effect for calculating ratio for data moving from Oracle database to non-Oracle database through gateway during INSERT, UPDATE, or DELETE.

HS_FDS_REMOTE_DB_CHARSET

Property	Description
Default Value	None
Range of values	Not applicable
Syntax	HS_FDS_REMOTE_DB_CHARSET

This parameter is valid only when HS_LANGUAGE is set to AL32UTF8 and the gateway runs on Windows. As more Oracle databases and non-Oracle databases use Unicode as database character sets, it is preferable to also run the gateway in Unicode character set. To do so, you must set HS_LANGUAGE=AL32UTF8. However, when the gateway runs on Windows, the Microsoft ODBC Driver Manager interface can exchange data only in the double-byte character set, UCS2. This results in extra ratio expansion of described buffer and column sizes. To compensate, the gateway can re-adjust the column size if HS_FDS_REMOTE_DB_CHARSET is set to the corresponding non-

Oracle database character set. For example, HS_FDS_REMOTE_DB_CHARSET=KO16KSC5601.

HS_FDS_SUPPORT_STATISTICS

Property	Description
Default Value	TRUE
Range of values	{TRUE|FALSE}
Syntax	HS_FDS_SUPPORT_STATISTICS= {TRUE|FALSE}

We gather statistics from the non-Oracle database by default. You can choose to disable the gathering of remote database statistics by setting the HS_FDS_SUPPORT_STATISTICS parameter to FALSE.

HS_FDS_SQLLEN_INTERPRETATION

Property	Description
Default Value	64
Range of values	{64|32}
Syntax	HS_FDS_SQLLEN_INTERPRETATION= {64|32}

This parameter is only valid for 64 bit platforms. ODBC standard specifies SQLLEN (of internal ODBC construct) being 64 bit on 64 bit platforms, but some ODBC driver managers and drivers violate this convention, and implement it as 32 bit. In order for the gateway to compensate their behavior, you need to specify HS_FDS_SQLLEN_INTERPRETATION=32 if you use these types of driver managers and driver.

HS_FDS_ARRAY_EXEC

Property	Description
Default Value	TRUE
Range of values	{TRUE|FALSE}
Syntax	HS_FDS_ARRAY_EXEC= {TRUE|FALSE}

If set to TRUE, the gateway will use array operations for insert, update, delete statements containing binds against the remote data source. The array size is determined by the value of the HS_FDS_FETCH_ROWS eter.

If set to FALSE, the gateway will not use array operations for insert, update, and delete statements. Instead, a single statement will be issued for every value.

Index

A

ALTER statement, *B-1*
Arithmetic operators, *B-3*

B

BYTE data type, *A-1*

C

Case rules, *2-4*
Case studies, *3-1*
case1.sql, *3-2*
CHAR data type, *A-1*
character sets
  Heterogeneous Services, *D-5*
COMMIT
  restrictions, *2-9*
Commit point site, *2-8*
Comparison operators, *B-3*
CONNECT BY clause, *2-10*
CREATE statement, *B-1*
Cursor loops
  restrictions, *2-9*

D

Data definition language, *B-1*
Data dictionary
  views, *C-2*
Data type
  BYTE, *A-1*
  CHAR, *A-1*
  conversion, *2-6*
  DATETIME, *A-1*
  DECIMAL, *A-1*
  FLOAT, *A-1*
  INTEGER, *A-1*
  LONG, *A-3*
  LONG RAW, *A-1*
  MONEY, *A-2*
  NCHAR, *A-2*
  NVARCHAR, *A-2*
Data type *(continued)*
  SERIAL, *A-2*
  SMALLFLOAT, *A-3*
  TEXT, *A-3*
  VARCHAR, *A-2*, *A-3*
  VARCHAR2, *A-2*, *A-3*
Date functions, *B-4*
DATETIME data type, *A-1*
DECIMAL data type, *A-1*
DELETE statement, *3-5*, *B-1*, *B-2*
  restrictions on use, *2-10*
Demo build SQL script, *3-2*
Demonstration files, *3-2*
Demonstration tables, *3-2*
Demonstration tables build SQL script, *3-2*
describe cache high water mark
  definition, *D-4*
DROP statement, *B-1*

E

Encrypted format login, *2-11*
Error messages
  error tracing, *D-10*
Errors
  ORA-02070, *2-8*

F

fetch array size, with HS_FDS_FETCH_ROWS, *D-11*
FLOAT data type, *A-1*

G

Gateway
  case studies, *3-1*
  pass-through feature, *2-1*
  supported functions, *B-1*
  supported SQL syntax, *B-1*
globalization support
  Heterogeneous Services, *D-4*
GRANT statement, *B-1*
Group functions, *B-3*

H

Heterogeneous Services
  defining maximum number of open cursors, D-6
  optimizing data transfer, D-6
  setting global name, D-4
  specifying cache high water mark, D-4
  tuning internal data buffering, D-7
  tuning LONG data transfer, D-6
Hexadecimal notation, 2-5
HS_DB_NAME initialization parameter, D-4
HS_DESCRIBE_CACHE_HWM initialization parameter, D-4
HS_FDS_CONNECT_INFO, D-9
HS_FDS_FETCH_ROWS parameter, D-11
HS_FDS_TRACE_LEVEL initialization parameter, D-10
  enabling agent tracing, D-2
HS_FDS_TRANSACTION_ISOLATIONparameter, D-10
HS_FDS_TRANSACTION_LOG initialization parameter, D-11
HS_IDLE_TIMEOUT initialization parameter, D-11
HS_KEEP_REMOTE_COLUMN_SIZE initialization parameter, D-12
HS_LANGUAGE initialization parameter, D-4
HS_LONG_PIECE_TRANSFER_SIZE initialization parameter, D-6
HS_NLS_LENGTH_SEMANTICS initialization parameter, D-12
HS_OPEN_CURSORS initialization parameter, D-6
HS_RPC_FETCH_REBLOCKING initialization parameter, D-6
HS_RPC_FETCH_SIZE initialization parameter, D-7
HS_TIME_ZONE initialization parameter, D-7

I

IFILE initialization parameter, D-8
Initialization parameter file
  customizing, D-1
INSERT statement, 3-5, B-1, B-2
  restrictions on use, 2-10
INTEGER data type, A-1
isolation level, HS_FDS_TRANSACTION_ISOLATION, D-10

K

Known restrictions, 2-7

L

Locking, database, 2-7
LONG data type, A-3
LONG RAW data type, A-1

M

MONEY data type, A-2

N

NCHAR data type, A-2
NVARCHAR data type, A-2

O

Objects
  valid characters, 2-4
Objects, naming rules, 2-4
ORA-02070, 2-8
Oracle functions
  TO_DATE, B-5

P

parameters
  gateway initialization file
    HS_FDS_FETCH_ROWS, D-11
    HS_FDS_TRANSACTION_ISOLATION, D-10
Pass-Through Feature, 3-5
Passing commands to database, 2-9
Pattern matching, B-4
PL/SQL, 2-12

R

RAW data type, 2-5
REAL data type, 2-9
remote
  HS_FDS_TRANSACTION_ISOLATION, D-10
ROLLBACK
  restrictions, 2-9
ROWID, 2-9, 2-10

S

savepoint support, 2-8
SELECT statement, 3-6, B-2, C-1
  use of NULL keyword, 2-10
SERIAL data type, A-2
Set operations, B-4

SMALLFLOAT data type, *2-9*, *A-3*
SQL
    statements,
        HS_FDS_TRANSACTION_ISOLATION, *D-10*
Stored procedures, *2-11*
String functions, *B-4*
SUM function, *3-4*

T

TEXT data type, *A-3*
TO_DATE function, *B-5*
transactional capability, *2-8*
transactional integrity, *2-8*
TRUNCATE statement, *B-1*
Two-phase commit, *2-8*

U

UPDATE statement, *3-5*, *3-6*, *B-2*
    restrictions on use, *2-10*

V

VARCHAR data type, *A-2*, *A-3*
VARCHAR2 data type, *A-2*, *A-3*

W

WHERE CURRENT OF clause, *2-9*