Altibase Tools & Utilities

Adapter for Oracle User's Manual

Release 6.1.1

February 4, 2013

ALTIBASE

Contents

Preface

About This Manual

This manual explains the Adapter for Oracle utility, whose purpose is to apply data that have been changed in ALTIBASE HDB to an Oracle database.

Audience

This manual has been prepared for the following ALTIBASE HDB users:

- database administrators

- application developers

- technical support workers

It is recommended that those reading this manual possess the following background knowledge:

- basic knowledge in the use of computers, operating systems, and operating system utilities

- experience in using relational databases and an understanding of database concepts

- computer programming experience

- experience in database server, operating system or network administration

Software Environment

This manual has been written assuming that ALTIBASE HDB 6 will be used as the database server.

Organization

This manual has been organized as follows:

- Chapter1: Introduction

    This chapter describes the concept of Adapter for Oracle and the mechanism by which data that have been changed in ALTIBASE HDB are written to an Oracle DB.

- Chapter2: Installation and Configuration

    This chapter explains how to install and configure Adapter for Oracle.

- Chapter3: Managing Adapter for Oracle

    This chapter explains to administrators how to manage Adapter for Oracle.

Documentation Conventions
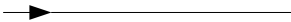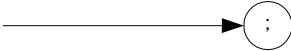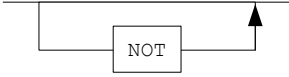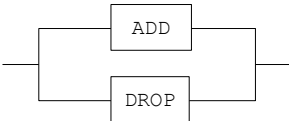
This section describes the conventions used in this manual. Understanding these conventions will make it easier to find information in this manual and other manuals in the series.
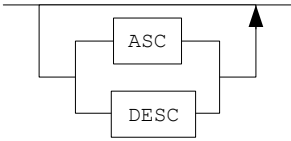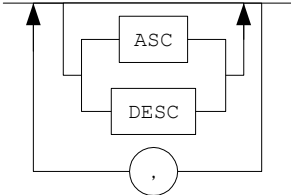
There are two sets of conventions:

- syntax diagram conventions

- sample code conventions

Syntax Diagram Conventions

This manual describes command syntax using diagrams composed of the following elements:

Elements	Meaning
Reserved word	Indicates the start of a command. If a syntactic element starts with an arrow, it is not a complete command.
⟶	Indicates that the command continues to the next line. If a syntactic element ends with this symbol, it is not a complete command.
⟶	Indicates that the command continues from the previous line. If a syntactic element starts with this symbol, it is not a complete command.
⟶ ;	Indicates the end of a statement.
___	Indicates a mandatory element.
NOT	Indicates an optional element.
ADD DROP	Indicates a mandatory element comprised of options. One, and only one, option must be specified.

Elements	Meaning
	Indicates an optional element comprised of options.
	Indicates an optional element in which multiple elements may be specified. A comma must precede all but the first element.

Sample Code Conventions

The code examples explain SQL statements, stored procedures, iSQL statements, and other command line syntax.

The following table describes the printing conventions used in the code examples.

Rules	Meaning	Example
[]	Indicates an optional item.	`VARCHAR [(size)] [[FIXED ]] VARIABLE]`
{}	Indicates a mandatory field for which one or more items must be selected.	`{ ENABLE	DISABLE	COMPILE }`
|	A delimiter between optional or mandatory arguments.	`{ ENABLE	DISABLE	COMPILE }` `[ENABLE	DISABLE	COMPILE]`
. . .	Indicates that the previous argument is repeated, or that sample code has been omitted.	`iSQL> select e_lastname from employees;` `E_LASTNAME` `-----------------------` `Moon` `Davenport` `Kobain` `.` `.` `.` `20 rows selected.`
Other Symbols	Symbols other than those shown above are part of the actual code.	`EXEC :p1 := 1;` `acc NUMBER(11,2);`

Rules	Meaning	Example
Italics	Statement elements in italics indicate variables and special values specified by the user.	`SELECT * FROM table_name;` `CONNECT userID/password;`
Lower case words	Indicate program elements set by the user, such as table names, column names, file names, etc.	`SELECT e_lastname FROM employees;`
Upper case words	Keywords and all elements provided by the system appear in upper case.	`DESC SYSTEM_.SYS_INDEX_;`

Related Documents

For more detailed information, please refer to the following documents:

- Administrator's Manual

- Replication Manual

- Log Analyzer User's Manual

- Utilities Manual

- Error Message Reference

Online Manual

Online versions of our manuals (PDF or HTML) are available from the Altibase Customer Support (http://support.altibase.com/).

Altibase Welcomes Your Opinions

Please feel free to send us your comments and suggestions regarding this manual. Your comments and suggestions are important to us, and may be used to improve future versions of the manual. When you send your feedback, please make sure to include the following information:

- The name and version of the manual you are using

- Your comments and suggestions regarding the manual

- Your full name, address, and phone number

In addition to suggestions, this address may also be used to report any errors or omissions discovered in the manual, which we will address promptly.

If you need immediate assistance with technical issues, please contact the Altibase Customer Support (http://support.altibase.com/).

We always appreciate your comments and suggestions.

1 Introduction

This chapter introduces the concept of Adapter for Oracle and how to apply data that have been modified in ALTIBASE HDB to an Oracle DB.

- About Adapter for Oracle

1.1 About Adapter for Oracle

ALTIBASE HDB Adapter for Oracle (hereinafter referred to as "oraAdapter") is a utility that enables data that have been modified in ALTIBASE HDB to be applied in an Oracle DB.

1.1.1 Structure and Concepts

In order for the user to copy data that have been modified in ALTIBASE HDB to an Oracle DB, it is first necessary to install ALTIBASE HDB, oraAdapter, and Oracle as shown in the following picture. The ALTIBASE HDB Log Analysis API (hereinafter referred to as "ALI") and the Oracle Call Interface (hereinafter referred to as "OCI") are integrated in oraAdapter. ALI is used to receive data that have been modified in ALTIBASE HDB, whereas the use of OCI enables the data to be written to an Oracle DB. For more detailed information about ALI, please refer to the Log Analyzer User's Manual.

The following picture illustrates how oraAdapter is used to apply data from ALTIBASE HDB to an Oracle DB.

Figure 1-1 The Structure of Adapter for Oracle
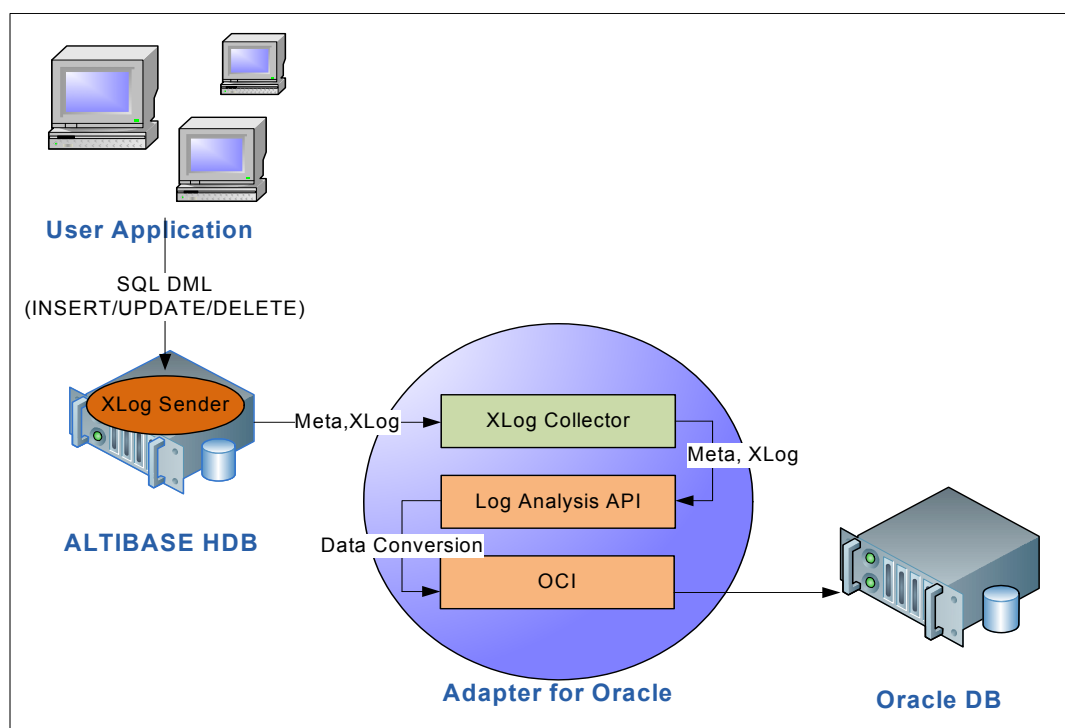
1. When the user creates or updates data, an XLog Sender that exists within ALTIBASE HDB creates XLogs and meta information and sends them to the XLog Collector. Note that meta information is exchanged only when handshaking takes place.

2. The XLog Collector, which exists within oraAdapter, uses ALI to provide XLogs and meta information to the user. If the call to ALI fails, a trace log will be written to the *trc* directory.

3. oraAdapter uses ALI to convert the acquired data so that they can be used with an Oracle DB.

4. The converted data are sent to the Oracle DB using OCI.

1.1.2 Terminology

1.1.2.1 XLog

An XLog is a logical log that is converted from a physical log. It is used to store the history of transactions involving DML (INSERT/UPDATE/DELETE) statements.

1.1.2.2 XLog Sender

The XLog Sender is the module that analyzes active logs to create XLogs and then passes them on to the XLog Collector.

The XLog Sender actively performs handshaking and XLog transmission.

1.1.2.3 XLog Collector

The XLog Collector is the module that receives meta data and XLogs from the XLog Sender.

The XLog Collector contains meta data, an XLog queue, a transaction table and an XLog pool.

1.1.2.4 Handshaking

Handshaking is the task of checking the protocol version and meta data before the XLog Sender sends XLogs to the XLog Collector.

1.1.2.5 Log Analysis API

The Log Analysis API provides XLogs and meta data that are used to interpret the XLogs.

1.1.2.6 OCI

The Oracle Call Interface (OCI) is a set of C-language APIs that provide an interface with an Oracle DB.

1.1.2.7 XLog Pool

The XLog Pool is memory that has been allocated for the storage of XLogs.

Introduction

1.1 About Adapter for Oracle

2 Installation and Configuration

In this chapter, the steps required to install Adapter for Oracle and make environment settings are described in detail.

- Pre-installation Tasks

- Installation

- Post-installation Tasks

- Configuration

- Properties

2.1 Pre-installation Tasks

In order to install and run Adapter for Oracle (oraAdapter), the following system requirements must be met. It is also necessary to make some environment settings for the system in question in order to ensure that oraAdapter runs properly.

For detailed information about system requirements, please contact the Altibase Customer Support Center (http://support.altibase.com).

2.1.1 OS

- AIX 5.3

- AIX 6.1

- LINUX

2.1.2 Database Versions

ALTIBASE HDB: version 5.5.1 or higher

Oracle Database: version 10g or higher (must be compatible with OCI)

2.1.3 Database Character Sets

This is the default character set for saving data. Setting ALTIBASE HDB and Oracle to use the same database character set can reduce the expense associated with data conversion, and thus it is recommended that they be set to use the same character set.

2.1.4 National Character Set

Data in a language that is not supported by the database character set can be saved using the NCHAR and NVARCHAR types. Just as with the database character set, setting ALTIBASE HDB and Oracle to use the same national character set can reduce the expense associated with data conversion, and thus it is recommended that they be set to use the same national character set.

2.1.5 Interface

Before installing oraAdapter, it is necessary to confirm that the Oracle Call Interface (hereinafter referred to as "OCI") has been installed. For more information on installing the OCI, please refer to the relevant Oracle documentation.
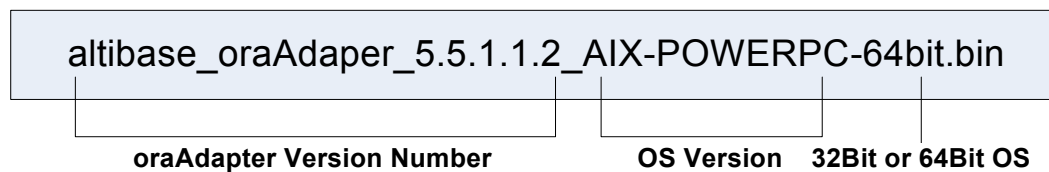
2.2 Installation

The oraAdapter installer can be executed in GUI mode if suitable display settings have been made. The description of oraAdapter installation in this manual will assume that the installation is being conducted in GUI mode. It is also possible to install oraAdapter in text mode if suitable display settings have not been made.

2.2.1 Installing oraAdapter

1. The first task is to obtain the version of the oraAdapter installer that is correct for the system on which it is to be run. The picture shows the naming convention for the oraAdapter installer.

    The oraAdapter version must be the same as the version of ALTIBASE HDB with which it will be run.

    altibase_oraAdaper_5.5.1.1.2_AIX-POWERPC-64bit.bin

    oraAdapter Version Number   OS Version   32Bit or 64Bit OS

2. When the installer is executed, the dialog appears. Verify that the version of oraAdapter to be installed is the correct version and click "Next".

3. The next dialog is for choosing the directory in which oraAdapter is to be installed. By default, a directory called "oraAdapter" will be created within the home directory of the user performing the installation.

4. To use oraAdapter, the ALTIBASE HDB Log Analyzer property settings must be made as follows. For more detailed information on properties, please refer to the section entitled "Properties" later in this chapter.

    *   ALA_SENDER_IP: This is the IP address of the server on which ALTIBASE HDB is installed. It is set to 127.0.0.1 by default, assuming that ALTIBASE HDB and oraAdapter are operating on the same server.
    *   ALA_RECEIVER_PORT: This is the number of the port on which oraAdapter listens. It can be set within the range from 1024 to 65535.
    *   ALA_REPLICATION_NAME: This is the name of a replication object that exists in ALTIBASE HDB.
    *   ALA_XLOG_POOL_SIZE: This is used to set the maximum size of the XLog pool. The default is 10,000 XLogs.
    *   ALA_SOCKET_TYPE: This is used to set the communication protocol that is used by ALA. oraAdapter supports the use of TCP/IP and the UNIX Domain Socket protocols.
    *   ALA_LOGGING_ACTIVE: This setting determines whether ALA will output trace logs. The default value is 1, which means that trace logs will be output.

5. In the next dialog, appropriate ALTIBASE HDB property settings must be made. For more detailed information on properties, please refer to the section entitled "Properties for Checking Constraints" later in this chapter.

- • ALTIBASE_USER : This is the name of the user account with which to access ALTIBASE HDB.
- • ALTIBASE_PASSWORD: This is the password corresponding to the user account with which ALTIBASE HDB will be accessed.
- • ALTIBASE_IP: This is the IP address of the server on which ALTIBASE HDB is installed. It is set to 127.0.0.1 by default, assuming that ALTIBASE HDB and oraAdapter are operating on the same server.
- • ALTIBASE_PORT: This is the number of the port at which ALTIBASE HDB listens. It can be set within the range from 1024 to 65535.

6. To use oraAdapter, appropriate Oracle DB property settings must be made. For more detailed information on properties, please refer to the section entitled "Properties for OCI" later in this chapter.

- • ORACLE_SERVER_ALIAS : This is used to specify an alias for accessing an Oracle DB that is already set in an Oracle Client. If this is not specified, access will be made to the Oracle DB that is set as the default host.
- • ORACLE_USER: This is the name of a user account with which to access the Oracle DB.
- • ORACLE_PASSWORD: This is the password corresponding to the user account with which to access the Oracle DB.
- • ORACLE_SKIP_INSERT : If you choose "YES", INSERT statements executed on ALTIBASE HDB will not be implemented on the Oracle DB.
- • ORACLE_SKIP_UPDATE: If you choose "YES", UPDATE statements executed on ALTIBASE HDB will not be implemented on the Oracle DB.
- • ORACLE_SKIP_DELETE : If you choose "YES", DELETE statements executed on ALTIBASE HDB will not be implemented on the Oracle DB.

7. In the next two dialogs, appropriate Oracle DB property settings must be made. For more detailed information on these properties, please refer to the section entitled "Properties for OCI" later in this chapter.

- • ORACLE_ASYNCHRONOUS_COMMIT: If you choose "YES", "asynchronous commit" will be used on the Oracle DB.
- • ORACLE_GROUP_COMMIT: If you choose "YES", "group commit" will be used on the Oracle DB.
- • ORACLE_ARRAY_DML_MAX_SIZE: "Array DML" means grouping multiple DML statements. This property sets the maximum number of DML statements that are grouped in this way. To disable Array DML, set this property to 1.
- • ORACLE_UPDATE_STATEMENT_CACHE_SIZE: This property is used to set the size of the cache in which prepared UPDATE statements are stored. If this property is set to 0, oraAdapter does not cache UPDATE statements.

8. Once all of the property settings pertaining to the use of oraAdapter have been made, they can be verified in the confirmation dialog box. Verify that all of the properties have been correctly set and click "Next".

9. After you have verified the property settings in the confirmation dialog box, you are ready to install oraAdapter. Click "Next" to continue.

10. While oraAdapter is being installed, the following two environment variables are set. Note that in order for the system to make use of the two new environment variables, it will be necessary to log out and log back in.

- ORA_ADAPTER_HOME: this environment variable is added with the oraAdapter home directory that was specified earlier during the installation process
- PATH: this environment variable is added with the value of "ORA_ADAPTER_HOME/bin".

11. The dialog box appears after oraAdapter has been successfully installed.

2.3 Post-installation Tasks

2.3.1 Setting Environment Variables

After oraAdapter has been installed, it is necessary to set environment variables, add a library path, and set the database and national character sets.

For more detailed information on setting environment variables, please refer to the section entitled Configuration later in this chapter.

- ORA_ADAPTER_HOME

  This is set automatically when oraAdapter is installed.

- Add a Library Path

  Add the path to the OCI library.

  ```
  $ export LIBPATH=$SIBPATH:$ORACLE_HOME/LIB
  ```

- NLS_LANG

  This is used to indicate the character set that is being used in ALTIBASE HDB.

2.3.2 Checking the Installation Directory

After oraAdapter installation is complete, verify that the bin, conf, msg, and trc directories have been created in the $ORA_ADAPTER_HOME directory. The structure and role of each directory is as follows:

- bin directory

  This directory contains the oraAdapter (Adapter for Oracle) and oaUtility (Adapter for Oracle Utility) executable files and some internally used files.

- conf directory

  The oraAdapter.conf file, in which the oraAdapter property settings are stored, is located in this directory.

- msg directory

  If any errors occur while oraAdapter is running, the messages in this directory will be used to write trace logs.

- trc directory

  The oraAdapter writes trace logs to files located in this directory.

2.4 Configuration

In order to use oraAdapter, it is necessary to configure environment variables.

2.4.1 ORA_ADAPTER_HOME

This is the directory in which oraAdapter was installed. This environment variable is set automatically during oraAdapter installation.

2.4.2 Library_Path

In order to use OCI to prepare data that have been changed in ALTIBASE HDB for use in an Oracle DB, the OCI library path must be added to the account that is used to execute oraAdapter.

The environment variable to which the library path must be added depends on the environment in which oraAdapter is executed.

In AIX, the OCI library path must be added to the LIBPATH environment variable.

2.4.3 NLS_LANG

The NLS_LANG environment variable is used with the OCI. This environment variable is used to indicate the character set of the strings that are inputted to the OCI. The OCI refers to this environment variable when converting data retrieved from ALTIBASE HDB to the Oracle database character set. Because oraAdapter receives data from ALTIBASE HDB, the NLS_LANG environment variable must be set to the Oracle DB character set that corresponds to the ALTIBASE HDB character set.

Note that if the character set of the Oracle DB is not the same as that of ALTIBASE HDB, performance will suffer as a result of the data conversion task performed by the OCI.

The syntax of the NLS_LANG environment variable is as follows.

NLS_LANG=KOREAN_KOREA.KO16MSWIN949

Language   Nation(Country)   Character Set
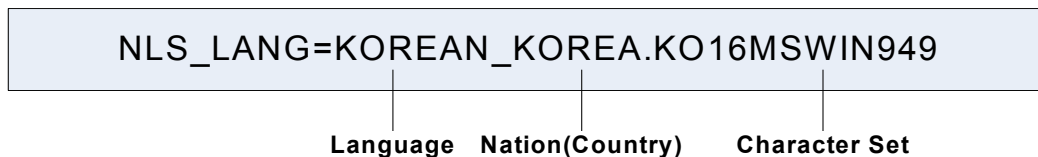
The following table shows the character sets that are supported for use with ALTIBASE HDB and the corresponding Oracle DB character sets and oraAdapter NLS_LANG environment variable settings.

Table 2-1 Database Character Set

ALTIBASE HDB	NLS_LANG	Oracle DB
US7ASCII	.US7ASCII	US7ASCII
KO16KSC5601	.KO16KSC5601	KO16KSC5601
MS949	.KO16MSWIN949	KO16MSWIN949

ALTIBASE HDB	NLS_LANG	Oracle DB
SHIFT-JIS	.JA16SJIS	JA16SJIS
EUC-JP	.JA16EUC	JA16EUC
GB231280	.ZHS16CGB231280	ZHS16CGB231280
BIG5	.ZHT16BIG5	ZHT16BIG5
UTF-8	.UTF8	UTF8

Table 2-2 National Character Set

ALTIBASE HDB	NLS_LANG	Oracle DB
UTF-8	.UTF8	UTF8
UTF-16	.AL16UTF16	AL16UTF16

2.5 Properties

The properties that are set within oraAdapter are used by the ALTIBASE HDB Log Analyzer and Oracle Call Interface, and are also necessary for making the settings that determine how oraAdapter will be executed.

The property file is oraAdapter.conf, which is located in the $ORA_ADAPTER_HOME/conf/ directory.

The properties that are used within oraAdapter can be classified as follows:

- Properties for ALA

- Properties for Checking Constraints

- Properties for OCI

- DML-Related Properties

2.5.1 Properties for ALA

The following properties must be set in order for oraAdapter to be able to use the ALTIBASE HDB Log Analyzer. For more detailed information about ALA properties, please refer to the *Log Analyzer User's Manual*.

2.5.1.1 ALA_SENDER_IP

This is the IP of the XLog Sender. Set it to the IP of the server on which ALTIBASE HDB is installed.

- Default Value : 127.0.0.1

2.5.1.2 ALA_RECEIVER_PORT

This is the port number at which the XLog Collector will listen to receive XLogs. Set it to the port number at which oraAdapter is listening.

- Range : 1024~65535

2.5.1.3 ALA_REPLICATION_NAME

This is the name of the replication object used as the XLog Sender. It is set to the name of a replication object created within ALTIBASE HDB.

2.5.1.4 ALA_SOCKET_TYPE

This is used to set the protocol that is used by the ALTIBASE HDB Log Analyzer. Adapter for Oracle supports TCP/IP and the Unix Domain Protocol. However, in order to use a UNIX domain socket, ALTIBASE HDB and Adapter for Oracle must be located on the same server.

- TCP : TCP/IP will be used(Default)

- UNIX : the Unix Domain Protocol will be used

2.5.1.5 ALA_XLOG_POOL_SIZE (Unit : Number）

This is used to set the maximum size of Adapter for Oracle's XLog Pool.

- Default Value : 10,000

- Range : 1~2147483647

2.5.1.6 ALA_LOGGING_ACTIVE

This is used to set whether the ALTIBASE HDB Log Analyzer will output Trace Logs.

- 0 : Trace logs are not output

- 1 : Trace logs are output（Default Value)

2.5.2 Properties for Checking Constraints

2.5.2.1 ALTIBASE_USER

This is used to specify the name of a user account with which to connect to ALTIBASE HDB.

2.5.2.2 ALTIBASE_PASSWORD

This is used to specify the password for the user account through which the connection to ALTIBASE HDB will be established.

2.5.2.3 ALTIBASE_IP

This is the IP address of the server on which ALTIBASE HDB is installed.

- Default Value : 127.0.0.1

2.5.2.4 ALTIBASE_PORT

This is the number of the port at which ALTIBASE HDB listens.

- Range : 1024 ~ 65535

2.5.3 Properties for OCI

The following properties must be set in order for Adapter for Oracle to be able to use the Oracle Call Interface（OCI).

2.5.3.1 ORACLE_SERVER_ALIAS

This property is used to specify an alias for an Oracle DB server that is set in the tnsnames.ora file, which contains information about servers to which Oracle Clients connect.

If this is not specified, connection will be made to the Oracle DB that is set as the default host.

If, for example, the tnsname.ora file contains the information shown below, ORACLE_SERVER_ALIAS should be set to orcl10g.

```
orcl10g =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = server1.us.oracle.com)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = orcl)
)
)
```

2.5.3.2 ORACLE_USER

This is used to specify the name of a user account with which to connect to the Oracle DB.

2.5.3.3 ORACLE_PASSWORD

This is used to specify the password for the user account through which the connection to the Oracle DB will be established.

2.5.3.4 ORACLE_ASYNCHRONOUS_COMMIT

This property determines whether or not to wait until commit logs have been written to disk.

- 0: This setting ensures that a commit message (i.e. a message indicating that the transaction has been committed) is returned to the client only after a commit log has been written to a persistent online redo log.

- 1: At this setting, a commit message is returned to the client regardless of whether the commit log has been completely written to disk. This is so-called "asynchronous commit", which can improve performance. Note that using asynchronous commit compromises durability in the interests of speed. Therefore, at this setting, when the Oracle database crashes, it may be necessary to synchronize the Oracle database with the ALTIBASE HDB database.

2.5.3.5 ORACLE_GROUP_COMMIT

If this property is enabled, the logs for multiple transactions that are occurring at the same time are accumulated in a buffer. After a certain amount of logs has amassed in the buffer, they are written to disk all at once. This is so-called "group commit", which allows redo information for multiple transactions to be written to disk in a single I/O operation.

Group Commit is suitable for environments in which transactions are frequently committed. The resultant improvement in performance comes at the cost of increased response times for individual transactions. It also introduces a problem whereby, if one of the transactions in the group cannot be committed, none of the transactions in the group are committed.

- 0: Group commit is not used.

- 1: Group commit is used.

2.5.3.6 ORACLE_ARRAY_DML_MAX_SIZE

"Array DML" means grouping multiple DML statements of the same kind. This realizes a performance improvement by reducing network costs.

- Default Value: 10

- Range: 1~32767

This property sets the maximum number of DML statements that are grouped in this way.

- It improves performance when the ORACLE_GROUP_COMMIT property is enabled.

- At present, this property only affects the INSERT and DELETE statements.

- To disable Array DML, set this property to 1.

2.5.3.7 ORACLE_UPDATE_STATEMENT_CACHE_SIZE

Statement caching improves performance by caching executable statements that are used repeatedly, such as in a loop or in a method that is called repeatedly.

- Default Value: 20

- Range: 0~4294967295（unit: number of statements)

This property is used to set the size of the cache in which prepared UPDATE statements are stored.

- Because the INSERT and DELETE statements already have their own cache, only UPDATE statements are stored in this cache.

- If this property is set to 0, oraAdapter does not cache UPDATE statements.

2.5.4 DML-Related Properties

These properties are used to set whether DML statements executed in ALTIBASE HDB will also be executed in the Oracle DB.

2.5.4.1 ORACLE_SKIP_INSERT

This property determines whether INSERT statements executed in ALTIBASE HDB are also executed in the Oracle DB. If it is set to 1 ("skip"), INSERT statements are skipped, meaning that they are not executed on the Oracle DB.

- 0: Do not skip the statements; that is, execute the statements normally.

- 1: Skip the statements.

2.5.4.2 ORACLE_SKIP_UPDATE

This property determines whether UPDATE statements executed in ALTIBASE HDB are also executed in the Oracle DB. If it is set to 1 ("skip"), UPDATE statements are skipped, meaning that they are not executed on the Oracle DB.

- 0: Do not skip the statements; that is, execute the statements normally.

- 1: Skip the statements.

2.5.4.3 ORACLE_SKIP_DELETE

This property determines whether DELETE statements executed in ALTIBASE HDB are also executed in the Oracle DB. If it is set to 1 ("skip"), DELETE statements are skipped, meaning that they are not executed on the Oracle DB.

- 0: Do not skip the statements; that is, execute the statements normally.

- 1: Skip the statements.

2.5 Properties

3 **Managing Adapter for Oracle**

This chapter explains how to start up and shut down oraAdapter and how to use the Adapter for Oracle Utility.

- Startup and Shutdown

- Data Types

- Adapter for Oracle Utility

- Command-Line Options

- FAQ

3.1 Startup and Shutdown

The following is an explanation of how to start up and shut down oraAdapter.

3.1.1 Startup

How to start up oraAdapter will now be explained step by step.

In order to use oraAdapter, ALTIBASE HDB and the Oracle DB must first both be running. Additionally, verify that the Adapter for Oracle property settings and environment variables have been set appropriately for the environment in which oraAdapter will be run.

If any environment variables or property settings are changed after oraAdapter has been started, it will be necessary to restart oraAdapter in order to apply the changes. For more information on setting environment variables, please refer to Post-installation Tasks in Chapter Two.

1. Check whether the ALTIBASE HDB REPLICATION_PORT_NO property[1] has been set to a port number that is actually available for use with replication. If it is necessary to change this property setting, it will also be necessary to restart ALTIBASE HDB.

2. Before starting oraAdapter, it is also necessary to configure the XLog Sender so that the ALTIBASE HDB Log Analyzer (ALA) can be used. The XLog Sender is used to send XLogs and Meta information from ALTIBASE HDB.

   In the following statement, an XLog Sender is created so that the data in table *t1*, which belongs to the sys user in ALTIBASE HDB, can be replicated to table *t2*, which belongs to the user *scott* in the Oracle DB.

   ```
   CREATE REPLICATION ala FOR ANALYSIS WITH '127.0.0.1', 25090
                        FROM sys.t1 TO scott.t2;
   ```

   CREATE REPLICATION ala **FOR ANALYSIS WITH** ´127.0.0.1´, 25090 **FROM** sys.t1 **TO** scott.t2;

ALA_REPLICATION_NAME	**Address of oraAdapter**	**Table Name in ALTIBASE HDB**
	ALA_RECEIVER_PORT	**Table Name in Oracle DB**

3. Now it is time to start oraAdapter. oraAdapter can be started either by executing it directly, or using the Adapter for Oracle Utility. For detailed information on how to start oraAdapter using the Adapter for Oracle Utility, please refer to the Adapter for Oracle Utility section later in this chapter.

   ```
   $ cd $ORA_ADAPTER_HOME/bin
   $ ./oraAdapter
   ```

4. Start the XLog Sender for the ALTIBASE HDB Log Analyzer. Any attempt to start the XLog Sender before oraAdapter has been started will fail.

   ```
   iSQL> ALTER REPLICATION ala START;
   Alter success.
   ```

---

1. REPLICATION_PORT_NO specifies the replication port number to use on a local server when the local server establishes a replication connection. For a complete description of this and other ALTIBASE HDB properties, please refer to the *General Reference*.

3.1.2 Shutdown

The process of shutting down oraAdapter includes the process of stopping the operation of the XLog Sender. If oraAdapter is shut down forcefully using the Adapter for Oracle Utility, it will be shut down successfully, but the ALTIBASE HDB XLog Sender will continue to attempt to connect to ora-Adapter.

```
iSQL> ALTER REPLICATION ala STOP;
Alter success.
```

3.2 Data Types

When data in ALTIBASE HDB are applied to an Oracle DB, the data types are converted as shown in the following table.

Table 3-1 Data Type Conversion

ALTIBASE HDB	Oracle DB	Additional Information for Oracle DB
FLOAT	NUMBER	
NUMERIC	NUMBER	
DOUBLE	NUMBER	BINARY_DOUBLE can also be used.
REAL	NUMBER	BINARY_FLOAT can also be used.
BIGINT	NUMBER	
INTEGER	NUMBER	
SMALLINT	NUMBER	
DATE	DATE	
CHAR	CHAR	
VARCHAR	VARCHAR2	
NCHAR	NCHAR	If the National Character Set on ALTIBASE HDB is set to UTF-16, the NCHAR and NVARCHAR types on the Oracle DB are twice as big as the corresponding types on ALTIBASE HDB.[1]
NVARCHAR	NVARCHAR2	

1. In ALTIBASE HDB, the size of the NCHAR and NVARCHAR types is based on the number of characters, whereas in Oracle DB the size of these types is determined based on the actual number of bytes.

3.2.1 Examples

The following example illustrates how the data types are converted when data in ALTIBASE HDB are applied to an Oracle DB, assuming that the National Character Set has been set to UTF-16.

Assuming that the data types in ALTIBASE HDB have been set as follows:

```
iSQL> CREATE TABLE T1(
               A1 INTEGER PRIMARY KEY,
               A2 CHAR(20),
               A3 VARCHAR(20),
               A4 NCHAR(20),
               A5 NVARCHAR(20)
                );
```

the data types in the Oracle DB would be set as shown below.

```
SQL> CREATE TABLE T1(
```

```
A1 NUMBER PRIMARY KEY,
A2 CHAR(20),
A3 VARCHAR2(20),
A4 NCHAR(40),
A5 NVARCHAR2(40)
);
```

```
A1 NUMBER PRIMARY KEY,
A2 CHAR(20),
```

3.3 Adapter for Oracle Utility

Adapter for Oracle Utility (oaUtility) is a script that runs oraAdapter as a daemon and check its status. This script runs in the bash shell.

The options supported by Adapter for Oracle Utility are as follows:

- oaUtility Start

- oaUtility Stop

- oaUtility Status

- oaUtility Check

3.3.1 oaUtility Start

3.3.1.1 Syntax

```
oaUtility {start}
```

3.3.1.2 Description

This is used to start oraAdapter as a daemon.

3.3.2 oaUtility Stop

3.3.2.1 Syntax

```
oaUtility {stop}
```

3.3.2.2 Description

This is used to forcefully terminate the Adapter for Oracle process.

When oraAdapter is forcefully shut down using this command, only Adapter for Oracle is shut down. The XLog Sender and Adapter for Oracle continually attempt to perform handshaking.

3.3.3 oaUtility Status

3.3.3.1 Syntax

```
oaUtility {status}
```

3.3.3.2 Description

This is used to check whether oraAdapter is running.

3.3.4 oaUtility Check

3.3.4.1 Syntax

```
oaUtility {check [alive|constraints]}
```

3.3.4.2 Description

If neither the alive nor constraints option is specified, oaUtility continually checks whether ora-Adapter is running, and if it has been shut down (regardless of whether it was shut down normally or forcibly), restarts it.

If either the alive or constraints option is specified, the corresponding task is performed only once, and then oaUtility shuts down (i.e. does NOT perform continual checking).

Specifying the alive option tells oaUtility to check whether oraAdapter is running.

Specifying the constraints option tells oaUtility to check whether each of the primary keys in the tables to be ported from ALTIBASE HDB to Oracle DB are defined consistently, i.e. on the basis of columns having the same name.

3.4 Command-Line Options

The following command-line options are supported for use with oraAdapter.

3.4.1 Syntax

```
oraAdapter [-v | -version]
```

3.4.2 Description

This option is used to output the version of ALTIBASE HDB with which oraAdapter was compiled.

3.4.3 Example

```
$./oraAdapter -v
ALTIBASE Adapter for Oracle version 5.5.1.1.2
...
```

3.5 FAQ

3.5.1 What do I have to do after I change an environment variable or a property?

Once oraAdapter has been started, if any environment variables or properties are changed, it will be necessary to restart oraAdapter in order to implement the changes.

3.5.2 What if the data are not successfully applied to the Oracle DB?

If oraAdapter fails to write a data item to the Oracle DB, it merely writes a log message and proceeds to the next data item. This log message will be left in a trace log file in the '$ORA_ADAPTER_HOME/ trc' directory.

3.5 FAQ

Index