

Oracle® Essbase

SQL Interface Guide

Release 11.1.2.2.000

ORACLE®

**ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM**

Essbase SQL Interface Guide, 11.1.2.2.000

Copyright © 1998, 2012, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

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

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, the following notice is applicable:

U.S. GOVERNMENT RIGHTS:

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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.

This software or hardware and documentation may provide access to or information on 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. 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.

Documentation Accessibility	5
Chapter 1. About SQL Interface	7
Understanding the SQL Interface Process	7
Preparing to Use SQL or Relational Data Sources	8
Chapter 2. Configuring Data Sources	9
About Configuring Data Sources	9
Configuring Data Sources on Windows	9
Configuring Data Sources on UNIX	10
Using Oracle Call Interface	10
Chapter 3. Preparing Multiple-Table Data Sources	13
Methods for Preparing Multiple-Table Data Sources	13
Access Privilege Requirements	13
Preferred Method—Creating One Table or View	13
Joining Tables During Data Loads	13
Chapter 4. Loading SQL Data	15
About Loading Data and Building Dimensions	15
Using Substitution Variables	15
Rules for Substitution Variables	16
Creating and Using Substitution Variables	16
Creating Rules Files and Selecting SQL Data Sources	16
Selecting SQL Data Sources	17
Creating SQL Queries (Optional)	17
Performing Multiple SQL Data Loads in Parallel to Aggregate Storage Databases	17
Chapter 5. Using Non-DataDirect Drivers	19
About Non-DataDirect Drivers	19
Creating Configuration Files for Non-DataDirect Drivers	19
Keywords and Values Used Within Configuration Files	19
Finding Driver Names on Windows	21
Finding Driver Names on UNIX	22
Configuring Non-DataDirect Drivers	22
Appendix A. Enabling Faster Data Loads from Teradata Data Sources	23
Using Teradata Data Sources	23
Installing Required Teradata Software	23
Configuring Teradata as a Data Source	24
Setting Up the Environment for Using Export Operator	26

Loading Teradata Data Using Teradata Parallel Transporter	27
Support for Unicode and Multibyte Character Sets	27
Index	29

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 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.

In This Chapter

Understanding the SQL Interface Process	7
Preparing to Use SQL or Relational Data Sources	8

Understanding the SQL Interface Process

You can use the SQL Interface feature to build dimensions and to load values from SQL and relational databases. For example, you can execute SQL statements that specify retrieval of only summary data.

You do not need SQL Interface for spreadsheet or text-file data sources that can be loaded using Oracle Essbase Administration Services, MaxL, or ESSCMD. See the *Oracle Essbase Database Administrator's Guide* and the *Oracle Essbase Technical Reference*.

With SQL Interface, you can load data from a Unicode-mode relational database to a Unicode-mode Oracle Essbase application. For information on the Essbase implementation of Unicode, see the *Oracle Essbase Database Administrator's Guide*.

SQL Interface works with Administration Services to retrieve data:

1. Using Administration Services, you write a SELECT statement in SQL.
2. SQL Interface passes the statement to a SQL or relational database server.

Note: As needed, SQL Interface converts SQL statements to requests appropriate to non-SQL databases.

3. Using the rules defined in the data-load rules file, SQL Interface interprets the records received from the database server. (For information on data-load rules files, see [Chapter 4](#), “Loading SQL Data.”)
4. SQL Interface loads the interpreted summary-level data into the database.

Preparing to Use SQL or Relational Data Sources

SQL Interface is installed during Essbase Server installation. See the *Oracle Enterprise Performance Management System Installation and Configuration Guide* for information about initial configuration tasks.

► To prepare for using SQL or relational data sources:

- 1 **Configure the ODBC driver, and point it to its data source.** See [Chapter 2, “Configuring Data Sources.”](#)
- 2 **If data is contained within multiple tables, perform an action:**
 - Before using SQL Interface, in the SQL database, create one table or view.
 - During the data load, join the tables by entering a SELECT statement in Administration Services.

See [“Methods for Preparing Multiple-Table Data Sources” on page 13](#) for instructions.
- 3 **Verify the data source connection by using Data Prep Editor, in Administration Services Console, to open the SQL source file.** See [Chapter 4, “Loading SQL Data.”](#)
- 4 **Create a rules file that tells SQL Interface how to interpret the SQL data that is to be used with the Essbase database.** See [Chapter 4, “Loading SQL Data.”](#)

After these steps are complete, you can load data or build dimensions; see [Chapter 4, “Loading SQL Data”](#).

2

Configuring Data Sources

In This Chapter

About Configuring Data Sources.....	9
Configuring Data Sources on Windows	9
Configuring Data Sources on UNIX	10
Using Oracle Call Interface	10

About Configuring Data Sources

Before using SQL Interface to access data, you must configure the operating system of each data source and the driver required for each data source.

The Essbase installation provides DataDirect ODBC drivers.

Note: The DataDirect ODBC drivers that connect to Oracle 11g databases are configured to enable multi-threaded connections and to disable uppercase conversion.

For detailed, driver-specific information on each DataDirect driver, see the *DataDirect Connect for ODBC Reference*. The location of this reference (typically within the `/EPM_ORACLE_HOME/common/ ... /books/odbc/odbceref/` directory), varies depending upon the platform.

To configure non-DataDirect ODBC drivers, or to change the default settings for DataDirect ODBC drivers, see [Chapter 5, “Using Non-DataDirect Drivers.”](#)

For a list of supported drivers, see *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Configuring Data Sources on Windows

On Windows, you use ODBC Administrator to configure data sources.

➤ To use ODBC Administrator to configure data sources:

- 1 Select **Start**, then **Administrative Tools**, and then **Data Sources (ODBC)**.
- 2 Select or add a data source, and enter the required information about the driver.

For detailed instructions, see the ODBC provider documentation.

Configuring Data Sources on UNIX

► To configure data sources on UNIX:

- 1 Open the `$ARBORPATH/bin/.odbc.ini` file and add a data source description.
- 2 If you add data sources or change driver products or data sources, you may need to edit the `.odbc.ini` file to update ODBC connection and configuration information, such as data source name and driver product name. Update instructions and requirements vary by platform.

Example: Updating `.odbc.ini` for DB2

Assuming this scenario:

- Connecting to a DB2 database named “tbc_data” that is on an isaix7 server
- Using an ODBC data source (named “db2data”) that invokes the DataDirect 6.1 Wire Protocol driver

To edit the `.odbc.ini` file, use the `vi` command and insert these example statements:

```
[ODBC Data Sources]
db2data=DB2 Source Data on isaix7
...
[db2data]
Driver=/vol1/Oracle/Middleware/EPMSysstem11R1/common/ODBC/Merant/6.1/lib/ARdb225.so
Database=tbc_data
IpAddress=isaix7
TcpPort=50000
```

Note: For detailed, driver-specific information on each DataDirect driver, see the *DataDirect Connect for ODBC Reference*. The location of this reference (typically within the `/EPM_ORACLE_HOME/common/ ... /books/odbc/odbceref/` directory), varies depending upon the platform.

Using Oracle Call Interface

You can use Oracle Call Interface (OCI) as an alternative to ODBC to significantly improve data load and dimension build performance. With this method, you use Data Prep Editor to specify an OCI connect identifier.

To use an Oracle OCI connect identifier, use the following syntax for the Data Source Name (DSN) identification:

```
host:port/Oracle_service_name
```

Following is an example OCI connect identifier where the host server name is “myserver,” the port number is 1521, and the Oracle Service Name is “orcl.us.oracle.com”:

```
myserver:1521/orcl.us.oracle.com
```

See also *Oracle Essbase Administration Services Online Help*.

On AIX systems, when you load SQL data using OCI, you must enable asynchronous I/O or the data load fails with this message:

```
Cannot get async process state. Essbase Error(1021104): Cannot load instant client
shared library [libociei.so].Make sure that the required binaries are present with
correct environment variables set.
```

► To enable asynchronous I/O on AIX:

1 Run this command to determine the state of the aio0 driver:

```
lsdev -C -l aio0
```

Example output:

```
aio0 Defined Asynchronous I/O
```

Defined indicates that the aio0 driver is installed on the system but is not available for applications to use. If the driver is not available for applications, change the state of the aio0 driver *Defined* to *Available*.

2 Run the cfgmgr AIX command:

```
cfgmgr -l aio0
```

3 To make the *Available* state permanent (across system reboots), issue the chdev AIX command:

```
chdev -l aio0 -P -a autoconfig='available'
```

You do *not* need to reboot the system to effectuate these changes.

Example output:

```
aio0 changed
```

4 Run this command to check the state of the aio0 driver:

```
lsdev -C -l aio0
```

Example output:

```
aio0 Available Asynchronous I/O
```


3

Preparing Multiple-Table Data Sources

In This Chapter

Methods for Preparing Multiple-Table Data Sources	13
Joining Tables During Data Loads	13

Methods for Preparing Multiple-Table Data Sources

- Before you use SQL Interface, in the SQL database, create one table or view.
- As you load data, join tables by entering a SELECT statement in Administration Services Console.

Access Privilege Requirements

For creating one table or view and for joining tables, you must have SELECT access privileges to the tables in which data is stored. For creating one table or view, you must have CREATE access privileges in the SQL database.

Preferred Method—Creating One Table or View

SQL database servers read from one table and maintain one view more efficiently than they process multiple-table SELECT statements. Therefore, creating one table or view before you use SQL Interface greatly reduces the processing time required by SQL servers.

Joining Tables During Data Loads

If you cannot obtain CREATE privileges, you must use Administration Services to join tables during the data load.

- To join tables during the data load:
- 1 Obtain **SELECT** access privileges to the tables in which relevant data is stored.
 - 2 In **Administration Services Console**, create a **SELECT** statement that joins the tables.
 - a. Identify the tables and columns that contain the data that you want to load into Essbase.
 - b. Select **File**, and then **Open SQL** to display **Open SQL Data Sources**.

See the *Oracle Essbase Administration Services Online Help*.

- c. Write a SELECT statement that joins the tables.

See [“Selecting SQL Data Sources” on page 17](#) and [“Creating SQL Queries \(Optional\)” on page 17](#).

Note: Essbase passes the SELECT statement to the database without verifying the syntax.

4

Loading SQL Data

In This Chapter

About Loading Data and Building Dimensions.....	15
Using Substitution Variables	15
Creating Rules Files and Selecting SQL Data Sources	16
Performing Multiple SQL Data Loads in Parallel to Aggregate Storage Databases	17

About Loading Data and Building Dimensions

After configuring one or more SQL data sources and preparing multiple-table data, you can use Oracle Essbase Administration Services to load data and build dimensions.

➤ To load data and build dimensions:

1 If you plan to use substitution variables, create them.

See [“Using Substitution Variables” on page 15](#).

2 Create rules files and select a data source.

See:

- [“Creating Rules Files and Selecting SQL Data Sources” on page 16](#)
- [“Performing Multiple SQL Data Loads in Parallel to Aggregate Storage Databases” on page 17](#)

3 Load data into the Essbase database.

See the *Oracle Essbase Administration Services Online Help*.

Using Substitution Variables

Using substitution variables in SQL strings and data source names enables you to use one rules file for multiple data sources. One substitution variable can apply to all applications and databases on an Essbase server or to a particular application or database.

You can also define substitution variables for data source names (DSNs) and specify in the rules file the substitution variable names.

Rules for Substitution Variables

- Use only valid and appropriate SQL values. Essbase does not validate values.
- Be especially careful with quotation marks (single and double). Different databases require different conventions.
- Because the ampersand (&) is the Essbase identifier for substitution variables, do not begin SQL operators in SELECT, FROM, or WHERE clauses with ampersands.

Creating and Using Substitution Variables

► To create and use substitution variables:

- 1 Using the instructions in the *Oracle Essbase Administration Services Online Help*, create the substitution variable.
- 2 As you edit the rule file, open the SQL data source by selecting **File**, then **Open SQL**.
See the *Oracle Essbase Administration Services Online Help*.
- 3 In the **Open SQL Data Sources** dialog box, perform an action:
 - To specify a substitution variable for the DSN, select **Substitution Variables**, and select a substitution variable.
 - To specify a substitution variable in the query, in **Select**, **From**, or **Where**, enter the substitution variable (with its preceding ampersand), instead of a “field=value” string.
- 4 Click **OK/Retrieve** to retrieve the data for the rules file.

Note: You must set the values for the substitution variables before you use the rules file for a data load or dimension build.

Creating Rules Files and Selecting SQL Data Sources

1. Create a data-load rules file; see the *Oracle Essbase Administration Services Online Help*.
Data-load and dimension-build rules are sets of operations that Essbase performs on data as the data is loaded into Essbase databases or used to build the dimensions of Essbase outlines. The operations are stored in rules files.
2. Select a SQL data source.
See [“Selecting SQL Data Sources” on page 17](#).
3. If you plan to create SQL queries in Essbase, see [“Creating SQL Queries \(Optional\)” on page 17](#).

Selecting SQL Data Sources

► To select SQL data sources:

- 1 In **Administration Services Console**, open **Data Prep Editor** or a rules file.
- 2 Select **File**, then **Open SQL**.
- 3 In **Select Database**, enter the names of the Essbase Server, application, and database, and click **OK**.
- 4 In **Open SQL Data Sources**, select the data source or the substitution variable, and enter required information.

See “Opening an SQL Database” in the *Oracle Essbase Administration Services Online Help*.
- 5 Click **OK/Retrieve**.
- 6 In **SQL Connect**, enter the user name and password for the source database, and click **OK**.

Facts about data source files:

- The data source file must be configured on the server computer.
- On UNIX platforms, the path for the SQL data source file is defined in the `.odbc.ini` file.
- On Windows, if the path for the SQL source file was not defined in ODBC Administrator, it can be entered in the Database box of the Define SQL dialog box.
- If a path is not defined, Essbase looks for the data source file in the directory from which Essbase Server is running.

Creating SQL Queries (Optional)

Instead of creating tables or views to select data for retrieval, you can write SELECT statements as you perform data loads.

Note: Creating SELECT statements in Essbase is usually slower than creating a table or view in the source database.

The SQL Statement box in the Open SQL Data Sources dialog box provides Select, From, and Where text boxes that help you write SQL queries. You can specify multiple data sources, filter the display of records, and specify how records displayed in Data Prep Editor are ordered and grouped.

Performing Multiple SQL Data Loads in Parallel to Aggregate Storage Databases

When loading SQL data into aggregate storage databases, you can use up to eight rules files to load data in parallel. Each rules file must use the same authentication information (SQL user name and password).

Essbase initializes multiple temporary aggregate storage data load buffers (one for each rules file), where data values are sorted and accumulated. When the data is fully loaded into the data load buffers, Essbase commits the contents of all buffers into the database in one operation, which is faster than committing buffers individually.

Note: This functionality is different than using the **import ... data to load_buffer with buffer_id** grammar to load data into a buffer, and then using the **import ... data from load_buffer with buffer_id** grammar to explicitly commit the buffer contents to the database. For more information on aggregate storage data load buffers, see the *Oracle Essbase Database Administrator's Guide*.

In MaxL, use the **import database** MaxL statement with the **using multiple rules_file** grammar. See the *Oracle Essbase Technical Reference*.

In the following example, SQL data is loaded from two rules files (`rule1.rul` and `rule2.rul`):

```
import database AsoSamp.Sample data
  connect as 'TBC identified by 'password'
  using multiple rules_file 'rule1' , 'rule2'
  to load_buffer_block starting with buffer_id 100
  on error write to "error.txt";
```

In specifying the list of rules files, use a comma-separated string of rules file names (excluding the `.rul` extension). The file name for rules files must not exceed eight bytes and the rules files must reside on Essbase Server.

In initializing a data load buffer for each rules file, Essbase uses the starting data load buffer ID you specify for the first rules file in the list (for example, ID 100 for `rule1`) and increments the ID number by one for each subsequent data load buffer (for example, ID 101 for `rule2`).

By default, SQL Interface disables parallel connections for the DataDirect ODBC drivers that are provided with Essbase. This feature requires parallel SQL connections; therefore, you must create a configuration file (`ARBORPATH/bin/esssql.cfg`) to change the default settings for the ODBC driver you are using. The following example of an `esssql.cfg` file for the SQL Server Wire Protocol driver provided with Essbase enables parallel SQL connections:

```
[
Description "SQL Server Wire Protocol"
DriverName ARMSSS
UpperCaseConnection 0
UserId 1
Password 1
Database 1
SingleConnection 0
IsQEDriver 0
]
```

You must restart Essbase Server for the change to take affect.

5

Using Non-DataDirect Drivers

In This Chapter

About Non-DataDirect Drivers	19
Creating Configuration Files for Non-DataDirect Drivers.....	19
Configuring Non-DataDirect Drivers	22

About Non-DataDirect Drivers

You must configure all non-DataDirect drivers (drivers other than the DataDirect drivers distributed with Essbase) for all data sources.

Some, but not all, non-DataDirect drivers are tested and supported for Essbase. For detailed information about qualified drivers and data sources, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

The information in the section also applies if you want to change the default settings for DataDirect ODBC drivers that are distributed with Essbase.

Creating Configuration Files for Non-DataDirect Drivers

You create a configuration file (*ARBORPATH/bin/esssql.cfg*) when you want to connect to a database using non-DataDirect drivers, or when you want to change the default settings for the DataDirect ODBC drivers that are distributed with Essbase.

- “Keywords and Values Used Within Configuration Files” on page 19
- “Finding Driver Names on Windows” on page 21
- “Finding Driver Names on UNIX” on page 22

Keywords and Values Used Within Configuration Files

The configuration file must contain the driver file names (*DriverName*), and an optional description (*Description*), the value for which must be enclosed in double quotation marks (the default value for *Description* is " "); it may contain additional keywords, the values for which are 0 or 1, as shown in [Table 1](#).

Keywords and values must be separated by at least one space, and the set of keywords and values for each driver must be enclosed within brackets ([]).

Different drivers may require additional values. See the driver documentation for specific information.

In the following example, the first entry for Oracle changes the default settings for the DataDirect drivers distributed with Essbase. The second entry is for Microsoft SQL Server, a non-DataDirect driver.

Sample esssql.cfg

```
[
Description "Oracle Wire Protocol"
DriverName ARORA
UpperCaseConnection 0
UserId 1
Password 1
Database 1
SingleConnection 0
IsQEDriver 1
]

[
Description "Microsoft SQL Server 32-bit"
DriverName SQLSRV32
UpperCaseConnection 0
UserId 1
Password 1
Database 1
SingleConnection 0
IsQEDriver 0
]

[
Description "Oracle BI Server"
DriverName libnqsodbc
UpperCaseConnection 0
UserId 1
Password 1
Database 1
SingleConnection 1
ConvertUTF16toUTF8 1
]
```

Defaults apply to values that are not specified. The defaults applied within configuration files differ from the Essbase default values that apply if no esssql .cfg file exists.

Table 1 Configuration File Keywords and Values for Non-DataDirect Drivers

Keyword	Value = 0	Value = 1
UserId	User ID not required (default)	User ID required
Password	Password not required (default)	Password required

Keyword	Value = 0	Value = 1
Database	Database name not required (default)	Database name required
Server	Server name not required (default)	Server name required
Application	Application name not required (default)	Application name required
Dictionary	Dictionary name not required (default)	Dictionary name required
Files	File name not required (default)	File name required
SingleConnection	Driver thread-safe—Multiple active connections permitted Note: Not recommended for non-Data Direct drivers, or for DataDirect drivers except for those used to connect to Oracle 11g databases, for which it is the default; may cause instability.	Driver not thread-safe—One active connection permitted The default and the recommendation for all DataDirect drivers except for those used to connect to Oracle 11g databases.
UpperCaseConnection	Driver case-sensitive—Connection information not converted (default) Tip: If the connection to the database server fails and the Application log shows an “invalid username/password; logon denied” message, check the case of the username and password in your database and compare it to what you are entering in Administration Services Console. To switch off case-sensitivity, change this value from 0 to 1.	Driver not case-sensitive—Connection information converted to uppercase
IsQEDriver	Driver a non-DataDirect driver (default)	Driver a DataDirect driver Note: You can specify configuration information for DataDirect drivers. For example, you can specify information for a version of a DataDirect driver that Essbase does not support.
ConvertUTF16toUTF8	No conversion of UTF16 data to UTF8 (this is the default).	Convert UTF16-encoded data from an Oracle BI data source to UTF8. This is required on UNIX for SQL data loads to Essbase from OBI.

Note: The DataDirect ODBC drivers that connect to Oracle 11g databases are configured to enable multi-threaded connections and to disable uppercase conversion. To enable multi-threaded connections for the SQL Server Wire Protocol driver, see [“Performing Multiple SQL Data Loads in Parallel to Aggregate Storage Databases”](#) on page 17.

Finding Driver Names on Windows

➤ To find driver names on Windows:

- 1 Using a method from [step 1](#) in [“Configuring Data Sources on Windows”](#) on page 9, start ODBC Administrator:

The **ODBC Data Source Administrator** dialog box opens.

Configured data sources are listed in **the User Data Sources** box. Drivers that are not properly configured but are listed in the User Data Sources box can be ignored.

- 2 Select the **Drivers** tab.
- 3 Obtain the file name of the preferred driver by scrolling to the right.

For example, the file name for the Microsoft Access Driver is `ODBCJT32.DLL`.

Finding Driver Names on UNIX

- To find driver names on UNIX, view the `.odbc.ini` file.

See [“Configuring Data Sources on UNIX” on page 10](#).

Configuring Non-DataDirect Drivers

Essbase recognizes the basic configuration information for DataDirect drivers, such as the name of the driver and whether the name and password are case-sensitive. You must provide configuration information for non-DataDirect drivers, or if you want to change the default settings for the DataDirect drivers that are distributed with Essbase.

- To provide configuration information:
 - 1 Create a configuration file (a text file) named `esssql.cfg`.
 - 2 Place the file in the `$ARBORPATH/bin` directory on Essbase Server.

In a default installation, `ARBORPATH` is: `EPM_ORACLE_INSTANCE/EssbaseServer/essbaseserver1`

Note: If you do not create a configuration file, Essbase uses default values that may prevent you from connecting to SQL databases.



Enabling Faster Data Loads from Teradata Data Sources

In This Appendix

Using Teradata Data Sources	23
Installing Required Teradata Software	23
Configuring Teradata as a Data Source	24
Setting Up the Environment for Using Export Operator	26
Loading Teradata Data Using Teradata Parallel Transporter	27
Support for Unicode and Multibyte Character Sets	27

Using Teradata Data Sources

You can use Teradata Parallel Transporter (TPT) from Teradata Tools and Utilities to significantly improve data load performance. With this method, ODBC is used to extract the database schema; then TPT retrieves the data.

Essbase supports Teradata Parallel Transporter (TPT) 12 with the Teradata 12 ODBC driver on the following platforms: 32-bit and 64-bit AIX, 32-bit HP-UX, 32-bit and 64-bit Red Hat Linux, 32-bit and 64-bit Solaris, 32-bit Windows, and 64-bit Windows AMD. TPT 12 can connect to Teradata V2R6.x and V12.x databases, not to Teradata V2R5.x databases. Consult your Teradata documentation for supported configurations.

For information about the versions of Teradata databases that Essbase supports as data sources, and supported Teradata ODBC drivers, see the *Oracle Enterprise Performance Management System Installation Start Here*.

Installing Required Teradata Software

The customer is responsible for having the correct Teradata license and ODBC version installed and configured on the Essbase Server computer. See the Teradata documentation for installation instructions.

- From Teradata Tools and Utilities, install Teradata Parallel Transporter Export Operator, Shared ICU Libraries for Teradata, Teradata GSS Client, and CLI. (For Linux installations, select libraries built by GCC 3.3.)
- Install the Teradata ODBC driver.

Configuring Teradata as a Data Source

► To configure Teradata as a data source:

1 Install Teradata drivers, which you must obtain from Teradata.

- Oracle Essbase Studio uses JDBC drivers. The JDBC Teradata driver must be installed on the computer on which Essbase Studio Server runs.

Essbase Studio uses the JDBC Teradata driver to deploy cubes in streaming mode.

To deploy cubes in non-streaming mode, the ODBC Teradata driver must be installed on the computer on which Essbase Server runs.
- Essbase uses ODBC drivers. The ODBC Teradata driver must be installed on the computer on which Essbase Server runs.

2 Stop Essbase from the Windows Services panel using the Oracle Process Manager and Notification Server (OPMN) service: EPM_epmsystem1.

3 Backup the OPMN configuration file (opmn.xml).

For example:

C:\Oracle\Middleware\user_projects\epmsystem1\config\OPMN\opmn\opmn.xml

4 Open the opmn.xml file in a text editor.

5 To properly load the Teradata drivers, the opmn.xml file must include a statement that points to the location of the Teradata libraries.

- a. Locate the following statement in the opmn.xml file:

```
<variable id="ESS_CSS_JVM_OPTION7" value="-  
Djava.util.logging.config.class=oracle.core.ojdl.logging.LoggingConfiguration"/>
```

- b. After this statement, add a statement similar to the following one:

```
<variable append="true" id="PATH" value="C:\Program Files\Teradata\Client\14.  
00\Shared ICU Libraries for Teradata\lib"/>
```

6 When using Teradata data sources with Essbase, and using OPMN to monitor and control the Essbase Agent process, you must update the opmn.xml file with variables for the operating system you are using.

Note: The absolute path value cannot contain spaces. The examples of absolute path values are based on a 64-bit machine configuration.

64-bit Windows

Add these variables:

- TWB_ROOT: Teradata root
- PATH: Teradata shared libraries
- PATH: Teradata client DLL libraries
- PATH: Teradata Call-Level Interface Version 2 routines

- PATH: Teradata message DLL libraries

64-bit Windows example:

```
<variable id="TWB_ROOT" value="C:\PROGRA~1\Teradata\Client\14.00"/>
<variable append="true" id="PATH" value="C:\PROGRA~1\Teradata\Client\14.
00\SHARED~1\lib"/>
<variable append="true" id="PATH" value="C:\PROGRA~1\Teradata\Client\14.
00\TERADA~1\bin64"/>
<variable append="true" id="PATH" value="C:\PROGRA~1\Teradata\Client\14.00\CLIV2"/>
<variable append="true" id="PATH" value="C:\PROGRA~1\Teradata\Client\14.
00\TERADA~1\msg64"/>
```

64-bit AIX

Add these variables:

- LIBPATH: Teradata ODBC libraries
- LIBPATH: Teradata shared libraries
- LIBPATH: ODBC components needed to load Teradata ODBC drivers
- LIBPATH: Teradata client libraries
- COPERR: Directory where the `errmsg.txt` file resides
- NLSPATH: Teradata message libraries

64-bit AIX example:

```
<variable append="true" id="LIBPATH" value="/opt/teradata/client/ODBC_64/lib"/>
<variable append="true" id="LIBPATH" value="/opt/teradata/client/13.10/tdicu/lib64"/
>
<variable append="true" id="LIBPATH" value="/usr/odbc/lib:/usr/odbc/drivers"/>
<variable append="true" id="LIBPATH" value="/usr/lib:/usr/teragss/aix-power/client/
lib"/>
<variable id=" COPERR" value="/usr/libperion/essbase"/>
<variable id="NLSPATH" value="/opt/teradata/client/13.10/odbc_32/msg/%N"/>
<variable append="true" id="NLSPATH" value="/usr/lib/nls/msg/%L/%N"/>
<variable append="true" id="NLSPATH" value="/usr/lib/nls/msg/%L/%N.cat"/>
```

64-bit LINUX

Add these variables:

- TWB_ROOT: Teradata root
- TD_ICU_DATA: Teradata shared libraries
- NLSPATH: Teradata ODBC message libraries
- COPERR: Directory where the `errmsg.txt` file resides
- COPLIB: Directory where the `libcliv2.so` library file resides
- LD_LIBRARY_PATH: Teradata libraries
- PATH: Teradata client directories

Note: The `errmsg.txt` and `libcliv2.so` files typically reside in the same directory. Therefore, the value for the `COPERR` and `COPLIB` variables is typically identical.

64-bit LINUX example:

```
<variable id="TWB_ROOT" value="/opt/teradata/client/13.10/tbuild"/>
<variable id="TD_ICU_DATA" value="</opt/teradata/client/13.10/tdicu/lib64>"/>
<variable id="NLSPATH" value="</opt/teradata/client/13.10/odbc_64/msg/%N >"/>
<variable append=true id="NLSPATH" value="/opt/teradata/client/13.10/tbuild/msg64/%N"/>
<variable id="COPERR" value="/usr/lib64"/>
<variable id="COPLIB" value="/usr/lib64"/>
<variable append=true id="LD_LIBRARY_PATH" value="/opt/teradata/client/13.10/tbuild/
lib64/>
<variable append=true id="LD_LIBRARY_PATH" value="/usr/lib64/>
<variable append=true id="PATH" value="/opt/teradata/client/13.10/tbuild/bin/>
<variable append=true id="PATH" value="/opt/teradata/client/13.10/tbuild/lib64/>
```

- 7 Save the `opmn.xml` file.
- 8 Start Essbase from the Windows Services panel using the Oracle Process Manager and Notification Server service (EPM_epmsystem1).
- 9 Verify the following:
 - Essbase: Use the Data Prep Editor in Administration Services Console to connect to a Teradata database using a DNS.
 - Oracle Essbase Studio: Perform a cube deployment in non-streaming mode, which uses the Teradata ODBC driver.

Setting Up the Environment for Using Export Operator

Follow the instructions in [Chapter 2, “Configuring Data Sources,”](#) and then perform these tasks:

- Add an entry to the hosts file for the Teradata database; for example:
172.27.24.181 tera2db tera2cop1
- Configure a system ODBC DSN for `$TELAPI$<tera>` where `<tera>` is the name of the Teradata data source; for example:
DSN = \$TELAPI\$tera2db
- For UNIX operating systems, ensure needed environment variable paths are defined in the appropriate location (the Windows installation automatically updates needed environment variables):
 - TD ODBC driver
 - CLIV2
 - TD GSS
 - Shared ICU
 - TPT export operator files
 - DataDirect ODBC driver

- In addition, in the appropriate path for the operating system, set the following variables for Teradata Parallel Transporter. (For details, see the “Code Samples” appendix in *Teradata Parallel Transporter Application Programming Interface Programmer Guide*); for example, for Solaris SPARC:
 - `export LD_LIBRARY_PATH = <library path>:$LD_LIBRARY_PATH`
`export LD_LIBRARY_PATH = /usr/tbuild/12.00.00/lib:$LD_LIBRARY_PATH`
 - `export NLSPATH = <directory path of the catalog>/%N:$NLSPATH`
`export NLSPATH = /usr/tbuild/12.00.00/msg/%N:$NLSPATH`
 - (If CLI is not installed in the default directory) `export COPERR = <directory location of errmsg.cat >`
`export COPERR = /usr/lib`

Loading Teradata Data Using Teradata Parallel Transporter

Follow the instructions in [Chapter 4, “Loading SQL Data.”](#) When you open the SQL data source, select the desired data source name with the prefix \$TELAPI\$ that you defined as the ODBC DSN. For the SQL statement, define a native Teradata query in the SQL SELECT, FROM, and WHERE statements. Do NOT include carriage returns or line feeds in these statements. Each entry must be in a single statement. See the relevant Teradata documentation for native Teradata SQL query rules.

Support for Unicode and Multibyte Character Sets

Teradata supports multibyte character set (MBCS) and Unicode text, which Essbase retrieves using TPTapi.

To use this functionality, perform these tasks:

- Verify that the client character set that Essbase uses is installed or enabled in the Teradata database.
- Make sure that the character set of the ODBC driver matches the character set that Oracle Essbase passes to TPTapi.

To do so, you should create the ODBC connection DSN with the character set name that matches that used by the \$ESSLANG variable, as shown in [Table 2](#).

Table 2 Supported Character Sets

Character Set	\$ESSLANG Variable Setting
Latin (covers almost all western languages)	Various
Japanese	KANJISJIS_05
Unicode	UTF8

Note: Essbase retrieves data in the supported character set; however, the SQL queries must be in English.

Index

Symbols

.odbc.ini file, DB2 example, [10](#)

A

Administration Services Console

joining tables, [13](#)

loading SQL data, [15](#)

SELECT statement, [13](#)

aggregate storage data load buffers, [18](#)

aggregate storage database, loading SQL data to, [17](#)

Application keyword, [21](#)

B

bin directory, [22](#)

building Essbase dimensions, [7](#)

C

CFG (.cfg) file, creating, [22](#)

choosing, a driver, [9](#)

configuration file, for non-DataDirect drivers, [19](#)

configuring

data sources

overview, [9](#)

non-DataDirect drivers, [22](#)

converting SQL statements, [7](#)

ConvertUTF16toUTF8 keyword, [21](#)

creating

data load rules files, [16](#)

driver configuration file, [19](#), [20](#), [22](#)

single table or view, [13](#)

D

data

load preparation, [13](#)

load preparation methods, [13](#)

SQL, mapping, [16](#)

data load rules files. *See* rules files.

data sources

paths, [17](#)

selecting, [17](#)

Database keyword, [21](#)

DataDirect drivers

changing default settings for, [19](#)

information on, [9](#), [10](#)

DB2, example .odbc.ini file, [10](#)

Description keyword, [19](#)

dialog box, ODBC Data Source Administrator, [21](#)

Dictionary keyword, [21](#)

dimensions, building, [7](#)

directories

bin, [22](#)

data sources, default, [17](#)

driver configuration file, [22](#)

driver configuration file

creating, [19](#), [20](#), [22](#)

default values, [20](#), [22](#)

in bin directory, [22](#)

keywords, [20](#)

required values, [19](#)

DriverName keyword, [19](#)

drivers

choosing in ODBC Administrator, [9](#)

description of, [19](#)

file names, [22](#)

naming, [19](#)

non-DataDirect, [19](#)

password, [22](#)

UNIX, names, [22](#)

E

Essbase

default values for drivers, [20](#), [22](#)

loading data into, [15](#)

- outline, mapping data to, [16](#)
- ESSBASEPATH, [22](#)
- esssql.cfg
 - configuring non-DataDirect drivers with, [22](#)
 - creating, [19](#)
 - example, [20](#)
 - keywords and values in, [20](#)
- examples, .odbc.ini files, [10](#)

F

- Files keyword, [21](#)
- files, CFG (configuration), [22](#)

I

- import database MaxL statement, [18](#)
- installed ODBC drivers, listed, [22](#)
- interpreting SQL records, [7](#)
- IsQEDriver keyword, [21](#)

J

- joining tables, [13](#)
- joining tables, in Administration Services console, [13](#)

K

- keywords, driver configuration file, [20](#)

L

- loading data
 - aggregate storage databases and, [17](#)
 - data load rules, [16](#)
 - from single table, [13](#)
 - in parallel, [17](#)
 - joining tables, [13](#)
 - overview, [8](#), [15](#)
 - preparing for, [13](#)
 - summary level data, [7](#)
 - using Administration Services, [15](#)

M

- mapping SQL data to an Essbase outline, [16](#)
- Microsoft Access, ODBC driver for, [22](#)
- multibyte character sets, support for, [27](#)
- multidimensional databases
 - building Essbase dimensions, [7](#)

- loading data into, [7](#)
- multiple rules files, using, [18](#)
- multiple tables
 - joining, [13](#)
 - loading from, [13](#)
 - preparing, [13](#)

N

- non-DataDirect drivers
 - configuring, [19](#)
 - tested, [19](#)

O

- ODBC Administrator, [9](#)
- ODBC drivers
 - file names, [22](#)
 - non-DataDirect, [19](#)
 - tested, [19](#)
 - UNIX, names, [22](#)
- ODBCJT32.DLL, [22](#)

P

- parallel SQL connections, [18](#)
- passing, SQL statements, [7](#)
- Password keyword, [20](#)
- passwords for driver, [22](#)
- paths
 - esssql.cfg, [22](#)
 - for data sources, [17](#)
- processing time, [13](#)

R

- relational data, loading:, [7](#). *See also* data.
- requesting non-SQL data, [7](#)
- rules files
 - creating, [16](#)
 - definition, [16](#)
 - for aggregate storage databases, [17](#)
 - for loading data in parallel, [17](#)
 - location of, [18](#)
 - maximum file name size, [18](#)
 - SQL records, [7](#)
- rules files, multiple, using, [18](#)

S

- SELECT statement
 - joining tables, [13](#)
 - processing time, [13](#)
- selecting SQL data sources, [17](#)
- Server keyword, [21](#)
- SingleConnection keyword, [21](#)
- spreadsheet data sources, [7](#)
- SQL data
 - joining tables, [13](#)
 - loading, [7](#)
 - mapping, [16](#)
 - selecting source, [17](#)
- SQL databases, single tables in, [13](#)
- SQL queries, [17](#)
- SQL server, [13](#)
- SQL statements, [7](#)
- summary data, loading, [7](#)

T

- tables
 - creating, [13](#)
 - efficiency, [13](#)
 - joining, [13](#)
 - loading from, [13](#)
 - SQL databases, [13](#)
- Teradata data source, using Teradata Export Operator, [23](#)
- tested non-DataDirect drivers, [19](#)
- text file data sources, [7](#)

U

- Unicode, support for, [7](#), [27](#)
- UNIX environments, odbc.ini file, [22](#)
- UpperCaseConnection keyword, [21](#)
- UserId keyword, [20](#)
- using multiple rules files, [18](#)

V

- views, [13](#)
 - creating, [13](#)
 - SQL databases, [13](#)

W

- Windows environments, configuring data sources, [9](#)

A B C D E F I J K L M N O P R S T U V W