IBM® DB2® OLAP Server™ and DB2 OLAP Starter Kit

Version 7
OLAP Integration Server Administrator’s Guide

P/N: SC27-0788-02

Printed in the U.S.A.

Contents

Chapter 4: Tables in OLAP Metadata Catalog

Chapter 5: Using Essbase Integration Services Shell

Chapter 6: Using the Administration Manager

Glossary

Appendix A: Essbase Integration Server Error Messages

Index

Preface

Purpose

This guide provides the information you need to configure and maintain Essbase® Integration Services and OLAP Metadata Catalog and to use Essbase Integration Services Shell on all supported platforms.

Audience

This guide is for Essbase Integration Services database administrators who are responsible for implementing, configuring, and managing Essbase Integration Services.

To use the information in this book, you need the following skills:

- Knowledge of where the data for your business resides (for example, in a relational database).
- A fundamental understanding of Microsoft Windows and basic Microsoft Windows terminology, such as dialog box, list box, and button.

  See Microsoft Windows documentation for more information on these terms.
- For UNIX system users, a fundamental understanding of UNIX system administration, such as directory navigation, environment variables, login scripts, and text file editing.

  See the documentation for the UNIX system you are using for more information.
- A fundamental understanding of database administration, such as database tables, columns and records, user management, operation of database client utilities, and executing SQL commands.

Document Structure

This document contains the following information:

Chapter 1, "Using Essbase Integration Server," describes how to start and stop the server and view the server error log.

Chapter 2, "Working with Users, Locks, and Permissions," describes how to view information about users, OLAP models and OLAP metaoutlines, and how to manage access to documents within an OLAP Metadata Catalog.

Chapter 3, "Troubleshooting ODBC and Connections," describes how to diagnose and resolve problems with database and software component connections, including Open Database Connectivity (ODBC) problems.

Chapter 4, "Tables in OLAP Metadata Catalog," describes the structure and content of the OLAP Metadata Catalog.

Chapter 5, "Using Essbase Integration Services Shell," describes the basic operation and commands for the Essbase Integration Services Shell.

Chapter 6, "Using the Administration Manager," describes how to access and use the Administration Manager

Appendix A, "Essbase Integration Server Error Messages," details the error messages generated by Essbase Integration Server.

Glossary, a list of key Essbase Integration Services terms and their definitions.

Index, a list of Essbase Integration Services terms and their page numbers. In the online version, select an index entry to view that page.

Related Documentation

The following documents contain additional related information:

Essbase Integration Services documents:

- *Essbase Integration Services Installation Guide,* for information about installation procedures, instructions for connecting relational data sources to OLAP Metadata Catalog, and information about Essbase Integration Services directories and files.

- *Essbase Integration Services OLAP Model User's Guide* (Release 2.0 version), for more information about basic multidimensional modeling concepts and about how to design, create, and maintain OLAP models based on a relational data source.
- *Essbase Integration Services OLAP Metaoutline User's Guide* (Release 2.0 version), for information about how to design, create, and maintain one or more metaoutlines based on an OLAP model.

Hyperion® Essbase® OLAP Server documents:

- *Essbase Database Administrator's Guide,* for more information about basic multidimensional concepts and how to design, create, and maintain an Essbase® database.
- *Essbase Installation Guide*, for information about how to install Essbase software components on all supported platforms.
- *Essbase SQL Interface Guide*, in online PDF format in the `\essbase\docs` directory, for information about how to use Essbase SQL Interface to load data from SQL, relational, and flat-file data sources into a Essbase OLAP Server database.
- *Essbase Spreadsheet Add-in User's Guide*, in online PDF format in the `\essbase\docs` directory, for information about how to use Essbase Spreadsheet Add-in with Microsoft Excel and Lotus 1-2-3 for Windows.
- The online *Quick Technical Reference*, in the `\essbase\docs` directory, for a complete listing and description of functions, calculation commands, report commands, and file settings.
- The online *API Reference* in the `\essbase\docs` directory, for a complete listing and description of functions available through the Essbase Application Programming Interface.

Online Help

➤ To access online help, click **Help** in any dialog box or select **Help > Help Topics**.

➤ To print an online help topic, display the topic and select **File > Print** or click the **Print** button.

Online Guides

The online guides are an electronic version of the printed documentation.

➤ To display an online guide, select **Start > Hyperion Solutions > Essbase Integration Services > Online Documentation** and select a title from the list of documentation.

Conventions

Table i shows the conventions used in this document.

Table i: Conventions Used in this Document

Item	Meaning
➤	Arrows indicate the beginning of a procedure, consisting of sequential steps.
Brackets [ ]	In examples, brackets indicate that the enclosed elements are optional.
Bold	Bold text indicates words or characters that you type exactly as they appear on the page. Bold in procedural steps highlights major user interface elements.
CAPITAL LETTERS	Capital letters denote commands and various IDs. (Example: LOADALL command)
`Example text`	Courier font indicates that the material shown is a code or syntax example.
Ctrl + 0	Keystroke combinations shown with the plus symbol (+) indicate that you should press the first key and hold it while you press the next key. Do not type the + symbol.
`Courier italics`	Courier italic text indicates a variable field in command syntax. Substitute a value in place of the variable shown in Courier italics.
Ellipses (…)	Ellipsis points indicate that text has been omitted from an example.
Italics	Italics in a product-related term indicates that the term is also included in the glossary of the book.

Table i: Conventions Used in this Document (Continued)

Item	Meaning
Mouse orientation	This document provides examples and procedures using a right-handed mouse. If you are using a left-handed mouse, adjust the procedures accordingly.
Menu options	Options in menus are shown in the following format: *Menu name* > *Menu command* > *Extended menu command* For example: View > Properties > Table
n, x	The variable *n* indicates that you must supply a generic number; the variable *x* indicates that you must supply a generic letter.

Note: The term right-click, used throughout this guide, means to click the secondary mouse button to open a pop-up menu.

Additional Support

In addition to using the documentation and online help, Hyperion offers the following additional support for product information:

Ordering Documentation

Additional copies of printed documentation may be ordered through your local customer support office.

Training Services

Hyperion offers a variety of training options, including instructor-led training, custom training, and multimedia training. This training covers all Hyperion applications and technologies and is geared to end users, administrators, and information systems (IS) professionals.

Instructor-led training is delivered in formats and in locations suited to Hyperion’s diverse, global customers. Hyperion Authorized Training Centers offer courses that they develop, as well as those developed by Hyperion. Custom training—training on the configured and tailored applications that employees use on the job—is another option to enhance user productivity and to ensure smooth day-to-day operations. Multimedia training—including computer-based training, Web-based training, and interactive distance learning—provides a cost-effective means of giving users a hands-on introduction to product features and functions. Computer-based training (CBT) provides high-quality interactive training at the user's convenience, regardless of location.

For more information about training, contact your local training services representative.

Consulting Services

Hyperion Consulting Services assists customers in maximizing the use of, and the return on investment in, Hyperion products. Experienced Hyperion consultants and Hyperion Alliance Partners assist organizations in tailoring solutions to their particular requirements, such as reporting, analysis, modeling, and planning. Specific services include implementation consulting, custom business solutions, data integration, and technical consulting. Additionally, Hyperion offers a variety of Services Packages and Reviews.

For more information about Consulting Services, Services Packages, and Reviews, as well as the services offered by Alliance Partners, contact your local consulting services representative.

Technical Support

Hyperion provides telephone and Web-based support to ensure that clients resolve product issues quickly and accurately. This support is available for all Hyperion products at no additional cost to clients with a current maintenance agreement.

When standard support does not meet specific requirements, a Hyperion support package that meets your needs usually can be designed. For more information, contact your local support office.

For information about technical support, visit the Hyperion Technical Support Web site located at `http://support.hyperion.com`.

Chapter

1

Using Essbase Integration Server

This chapter describes Essbase Integration Server, how to start it, how to shut it down, and how to view its log file. This chapter contains the following topics:

- "Introducing Essbase Integration Server" on page 1-1
- "Starting Essbase Integration Server" on page 1-2
- "Shutting Down Essbase Integration Server" on page 1-7
- "Viewing the Essbase Integration Server Log File" on page 1-7

Introducing Essbase Integration Server

Essbase Integration Server is a multithreaded server that serves as the bridge between the relational data source, OLAP Metadata Catalog, and Essbase OLAP Server. Essbase Integration Server performs the following tasks:

- Retrieves information about OLAP models and metaoutlines from OLAP Metadata Catalog
- Generates SQL statements
- Retrieves data from the external data source
- Loads members and data into the Essbase database

Essbase Integration Server must be running if you want to use Essbase Integration Services Console (the graphical user interface) or Essbase Integration Services Shell (the command line interface).

Starting Essbase Integration Server

Before you start Essbase Integration Server, make sure that the following programs are running. These programs can be on network server computers and do not need to be on your local computer to be available to Essbase Integration Server.

- A relational database management system (RDBMS) with the relational data source that contains the OLAP Metadata Catalog where the metadata is stored.
- An RDBMS that contains the relational data source you want to use to create an OLAP model or a metaoutline.

Essbase OLAP Server also must be running if you need to create, change, or load data into an Essbase database, or if you want to preview an Essbase database outline.

➤ To start Essbase Integration Server, use either of the following methods:

- From the Windows **Start** menu, select **Programs > Hyperion Solutions > Essbase Integration Services > Essbase Integration Server**.
- Type the executable name on the command line; for example, type

  ```
  olapisvr
  ```

  Note: If Essbase Integration Server does not start when `olapisvr` is executed from the command line, the operating system path may not be updated correctly. For more information, see the *Essbase Integration Services Installation Guide*.

When starting Essbase Integration Server from the command line, use the switches specified in Table 1-1 if you want to change the default settings:

Table 1-1: Essbase Integration Server Switches

Switch	Description
-X	To view a list of available switches for starting Essbase Integration Server, type **`-?`** when you start Essbase Integration Server; for example, type **`olapisvr -?`**
-B	To set the buffer size that Essbase Integration Server uses when transferring rows from the data source to the Essbase database, type **`-B`*`numberOfRows`*** when you start Essbase Integration Server. The default number is 100 rows. To maximize performance, estimate the size in bytes of the rows being loaded from the relational data source. Divide 64,000 by that estimated size to determine the optimal number of rows to buffer. The goal is to generate blocks of data that are as close to 64 KB as possible, without exceeding that amount. If the size of a block of rows exceeds the 64 KB threshold, the load will fail. For example, to set the number to 150, type **`olapisvr -B150`**

Table 1-1: Essbase Integration Server Switches (Continued)

Switch	Description
-C	To set the number of records that Essbase Integration Server commits to Essbase during a data load, type ***-Cnumber of records*** when you start Essbase Integration Server. This setting specifies the number of records to be committed to Essbase after all records are sent. The default setting is all records. The recommended setting for testing purposes is 5,000 to 10,000 records. After you have completed data load testing, shut down the server to reset the number of records to the default setting of all records. Setting a smaller number of records during testing, such as after every 5,000 or 10,000 records have been transferred to Essbase, enables you to see data load errors incrementally. Viewing the errors provides you an opportunity to correct data source problems before continuing and waiting for an entire data load to complete before you can take any corrective actions. Resetting the number of records to the default of all records results in faster performance, but error records are viewable only at the completion of the data load. There is no limit to the number of records that Essbase Integration Server can transfer to Essbase during a data load. Essbase can utilize all available memory resources allocated to it. However, if you experience memory problems during a data load, set -C to a smaller value. For example, to set the number of records that Essbase Integration Server incrementally transfers to Essbase during a data load, type **`-C5000`** **Note:** This setting does not apply to member loads.
-E	To give the log file a different name, type ***-Elog_file_name*** when you start Essbase Integration Server. The `.log` extension is added automatically. The default name and location is *ISHOME*`\bin\olapisvr.log`. For example, to name the log file `myserver.log` and put it in the `temp` directory, type **`olapisvr -Ec:\temp\myserver`**

Table 1-1: Essbase Integration Server Switches (Continued)

Switch	Description
-L	To set the level of detail of the messages that Essbase Integration Server logs, type ***-Llevel*** when you start Essbase Integration Server. The default level is 2. For example, to use the highest level of logging, type `olapisvr -L0` You can specify the following levels: 0 to log all debug messages and the ODBC SQL generated by Essbase Integration Server 2 to log all informational messages and some of the ODBC SQL generated by Essbase Integration Server 3 to log all warnings 4 to log all non-critical errors 7 to log all critical errors where Essbase Integration Server terminates the command 8 to log all critical server errors The recommended level settings are 2 or 3.
-M	To set the Essbase message database file to a different name, type **-M*database_file_name*** when you start Essbase Integration Server. The default name is *ARBORPATH*\bin\essbase.mdb. For example, to set the Essbase message database file name to essbase6\bin\essbase.mdb, type `olapisvr -Mc:\essbase6\bin\essbase.mdb`

Table 1-1: Essbase Integration Server Switches (Continued)

Switch	Description
-N	To specify the number of threads for executing SQL, retrieving data, transforming it, and sending it to Essbase during a data load, type **-N*number_of_threads*** when you start Essbase Integration Server. The default setting is 1 thread. The number of threads allocated to the Essbase Integration Server data load optimization process is controlled by the -N switch. For optimal data load performance, a thread can be allocated to each task in the data load process: executing the SQL statements, retrieving data, transforming data, and transferring data to Essbase. For example, to set the number of threads to 3, type `olapisvr -N3` The recommended setting is eight times the number of processors in the computer containing the server component of Essbase Integration Services.
-P	To set the TCP port number at which Essbase Integration Server listens, type **-P*portNumber*** when you start Essbase Integration Server. This switch is for debugging purposes only. Do not use this switch to set the port number. If you set the port number to a value other than the default, Essbase Integration Services Console will be unable to connect to Essbase Integration Server. The default port number is 3388. To access a port other than 3388, use olapicmd and specify **-P**, using the same value as specified for olapisvr.
-T	To set the number of network listeners that Essbase Integration Server starts, type **-T*number*** when you start Essbase Integration Server. The default number is 10. The recommended setting is 10. Network listeners receive requests from Essbase Integration Services Console. The server automatically adds and subtracts listeners as needed, so the number of listeners set at startup does not limit the number of users that can connect to an Essbase Integration Server. For example, to set the number to 17, type `olapisvr -T17`

Shutting Down Essbase Integration Server

➤ When you finish using Essbase Integration Server, shut it down by using either of the following methods:

- On Windows NT and Windows 2000 systems, close the Essbase Integration Server window.
- Issue the SHUTDOWN command in the Essbase Integration Services Shell.

  For more information, see "Using Essbase Integration Services Shell Commands" on page 5-9.

You must be logged on to Essbase Integration Server before you can shut it down.

Note: Always close Essbase Integration Services Console before shutting down Essbase Integration Server.

Viewing the Essbase Integration Server Log File

By default, the Essbase Integration Server log file is located in the same directory as Essbase Integration Server and is named `olapisvr.log`. You can view it by using any text editor. To create the log file with a different location or name, use the -E switch. For more information, see Table 1-1.

The following items are logged in the Essbase Integration Server log file, depending on the logging level you set when you start the server:

- All error messages
- The SQL statements that Essbase Integration Server generates to access the relational data source
- The Essbase Integration Services Shell commands used to perform an action
- Connection information
- Notification of the success or failure of a load
- The name, if applicable, of the metaoutline being run
- Other informational messages

For more information on setting the logging level, see Table 1-1.

Note: Essbase error message text is located in files in `ISHOME\esslib\bin\message.txt` on Windows systems.

Chapter 2

Working with Users, Locks, and Permissions

This chapter describes how to view information about users, OLAP models, and metaoutlines in OLAP Metadata Catalog. It also describes how to clear locks and change permissions on OLAP models and metaoutlines.

This chapter contains the following topics:

For more information about OLAP Metadata Catalog, see Chapter 4, "Tables in OLAP Metadata Catalog."

Working with Users

When working with Essbase Integration Services, you must manage access for three types of users:

- Essbase Integration Services users.

  These users can access OLAP models and metaoutlines stored in OLAP Metadata Catalogs. Essbase Integration Services users are the database users defined in the relational database that contains the OLAP Metadata Catalog you are using.

- Essbase users.

  These users can access the Essbase database that you create from a metaoutline.

- Source database users.

  Source database users are defined in the relational database management system (RDBMS) that manages the source database. These users can access the tables and views in the relational source database.

Use the tools provided with each relational database to manage each set of users. Consult the relational database documentation if you are not sure how to perform any of the following tasks:

- Create new users
- View a list of available users
- Change the permissions for users
- Delete current users
- Disconnect users

Working with Locks

You can use either of two methods for opening an OLAP model or metaoutline in Essbase Integration Services Console: standard access mode (the default mode) and exclusive access mode. The locks that are established by these different modes are specific to Essbase Integration Services and are completely separate from the locking performed by the RDBMS.

Tip: If you plan only to view an OLAP model or metaoutline, use standard access mode. If you plan to edit an OLAP model or metaoutline, use exclusive access mode.

Using Standard Access Mode

When you open an OLAP model in standard access mode, Essbase Integration Services gives you a read lock on that OLAP model. When you open a metaoutline in standard access mode, Essbase Integration Services gives you a read lock on that metaoutline and a read lock on the OLAP model on which the metaoutline is based. Multiple users can have read locks on the same OLAP model or metaoutline at the same time.

When you save an OLAP model or metaoutline in standard access mode, Essbase Integration Services upgrades the read lock to a write lock for the duration of the save. After Essbase Integration Services saves the changes to the OLAP model or metaoutline, the lock reverts back to a read lock. However, if other users have the OLAP model or metaoutline open, your changes are not updated for the other users until they reopen the document. In addition, if other users have the OLAP model or metaoutline open at the same time as you do, they can edit and save it, thereby overwriting your changes. Using exclusive access mode prevents users from overwriting your changes. For more information, see "Using Exclusive Access Mode" on page 2-3.

➤ If you have opened and edited an OLAP model or metaoutline in standard access mode, and are concerned that other users may overwrite your changes, perform one of the following steps:

- Save the OLAP model or metaoutline using a different name. For more information, see the *Essbase Integration Services OLAP Model User's Guide* and the *Essbase Integration Services OLAP Metaoutline User's Guide*.
- If you have not yet started editing the document or have entered only a few changes, close the OLAP model or metaoutline; then reopen it using exclusive access mode and reapply the changes.
- Contact the users with read locks and ask them to close the OLAP model or metaoutline. To view a list of users with read locks, see "Viewing Essbase Integration Services Users with Locks" on page 2-4.

Using Exclusive Access Mode

Opening an OLAP model or metaoutline in exclusive access mode eliminates the risk of other users overwriting your changes. When you open an OLAP model in exclusive access mode, Essbase Integration Services gives you a write lock on that OLAP model that remains in effect until you close the model. When you open a metaoutline in exclusive access mode, Essbase Integration Services gives you a write lock on that metaoutline and a read lock on the OLAP model on which the metaoutline is based that remains in effect until you close the metaoutline. While you have an OLAP model or metaoutline open in exclusive access mode, other users can open and view the documents, but they cannot save them.

Correcting Problems with Locks

Due to the complex nature of Essbase Integration Services and the other software components with which it interacts, you may experience problems that result in locks not being released from OLAP models or metaoutlines. Unreleased locks can result in an inability to save, even if no other users are on the system. Unreleased locks can also prevent you from opening an OLAP model or metaoutline.

If you suspect that unreleased locks remain from previous sessions, delete the locks by selecting Servers > OLAP Metadata Catalog > Delete Locks. Deleting locks from the menu removes only your locks; it does not delete locks held by other users.

If you suspect that unreleased locks remain from other user sessions, check to see which users have locks. For more information, see "Viewing Essbase Integration Services Users with Locks" on page 2-4. If you are convinced that some or all of the locks are from terminated user sessions, delete them. For more information, see "Deleting Locks for Essbase Integration Services Users" on page 2-6.

Viewing Essbase Integration Services Users with Locks

The CB_CONCURRENCY table in OLAP Metadata Catalog contains information about the users who access OLAP models and metaoutlines. This table contains the following columns:

- CB_USER_NAME: the name of the user accessing the OLAP model or metaoutline; for example, `sa`.
- CB_USER_SESSION_ID: a system-generated reference number that uniquely identifies an editing session
- CB_OBJECT_ID: an OLAP model or metaoutline number
- CB_OBJECT_TYPE: a type reference that indicates whether the user is accessing an OLAP model or a metaoutline
  - The number 1 represents an OLAP model.
  - The number 2 represents a metaoutline.
- CB_OBJECT_LOCK: a lock reference number that indicates whether the user has a read or write lock
  - The number 1 represents a read lock.
  - The number 2 represents a write lock.
- CB_LOCK_TIMESTAMP: the date and time that the user acquired the lock

➤ To determine which users are accessing a specific OLAP model or metaoutline:

1. Enter the following SELECT statement in the OLAP Metadata Catalog database, using the tools for the relational database you are using:

```
SELECT CB_USER_NAME, CB_OBJECT_ID, CB_OBJECT_TYPE,
       CB_OBJECT_LOCK, CB_LOCK_TIMESTAMP
FROM CB_CONCURRENCY
```

2. View the results.

   In the following example, the sa user is the only user currently accessing an OLAP model or metaoutline. The sa user has two read locks on one OLAP model and one read lock each on two different metaoutlines.

```
CB_USER_NAME CB_OBJECT_ID CB_OBJECT_TYPE CB_OBJECT_LOCK CB_LOCK_TIMESTAMP
------------ ------------ -------------- -------------- -----------------
sa           889844639        1               1         Apr 9 2001 4:43PM
sa           889845263        2               1         Apr 9 2001 4:43PM
sa           889844639        1               1         Apr 9 2001 5:20PM
sa           892167813        2               1         Apr 9 2001 5:20PM
                                                      (4 row(s) affected)
```

   You can determine the following information from the sample CB_CONCURRENCY table shown in the preceding list:

   - The first row of the results tells you that the sa user (CB_USER_NAME = sa) has a read lock (CB_OBJECT_LOCK = 1) on an OLAP model (CB_OBJECT_TYPE = 1) with an ID of 889844639.
   - The second row of the results tells you that the sa user (CB_USER_NAME = sa) has a read lock (CB_OBJECT_LOCK = 1) on a metaoutline (CB_OBJECT_TYPE = 2) with an ID of 889845263.
   - The third row of the results tells you that the sa user (CB_USER_NAME = sa) has a read lock (CB_OBJECT_LOCK = 1) on an OLAP model (CB_OBJECT_TYPE = 1) with an ID of 889844639.
   - The fourth row of the results tells you that the sa user (CB_USER_NAME = sa) has a read lock (CB_OBJECT_LOCK = 1) on a metaoutline (CB_OBJECT_TYPE = 2) with an ID of 892167813.

   When you open a metaoutline, you receive a read lock on the metaoutline and on the OLAP model on which it is based; therefore, you can assume that the sa user is working on two different metaoutlines based on the same OLAP model.

Deleting Locks for Essbase Integration Services Users

If you are certain that the other users who have read or write locks on an OLAP model or a metaoutline that you want to save do not need their locks, delete their locks from the CB_CONCURRENCY table.

CAUTION: Make sure the users do not need their locks before you delete the locks.

➤ To delete read and write locks on OLAP models or metaoutlines:

1. Determine which users have locks.

   See "Viewing Essbase Integration Services Users with Locks" on page 2-4.

2. Delete the rows containing the unwanted locks.

   For example, to delete all locks held by the `sa` user, issue the following DELETE statement in the OLAP Metadata Catalog database, using the tools for the relational database:

   ```
   DELETE FROM CB_CONCURRENCY WHERE CB_USER_NAME = 'sa'
   ```

Working with Permissions

When you save an OLAP model or metaoutline for the first time, you determine what read or write permissions other users have. Essbase Integration Services supports the following kinds of permissions:

- Allow read/write access for other users. This setting permits all other users to read and write to the OLAP model or metaoutline. This setting is the default.
- Allow read access for other users. This setting permits all other users to read but not write to the OLAP model or metaoutline; that is, other users cannot save changes to the OLAP model or metaoutline.
- Disallow all access for other users. This setting denies all other users read or write permission to the OLAP model or metaoutline. You are the only user who can read or write to it.

➤ To change the permissions of an OLAP model or metaoutline, take one of the following actions:

- Change the OLAP model properties in Essbase Integration Services Console. For more information, see the *Essbase Integration Services OLAP Model User's Guide*.
- Save the metaoutline with a different name by selecting **File > Save As**. For more information, see the *Essbase Integration Services OLAP Metaoutline User's Guide*.
- To edit tables containing the permissions for the OLAP model or metaoutline, use the tools provided with the relational database that contains OLAP Metadata Catalog. For more information on the tables to edit, see "Viewing Permissions for OLAP Models" on page 2-7 or "Viewing Permissions for Metaoutlines" on page 2-9.

Viewing Permissions for OLAP Models

Information about the permissions that are set for OLAP models and metaoutlines is stored in OLAP Metadata Catalog. View this information by selecting the appropriate columns from tables in OLAP Metadata Catalog.

The OM_INFO table in OLAP Metadata Catalog contains information about OLAP models, including the following columns which are relevant to permissions:

- MODEL_ID: a system-generated reference number.
- MODEL_NAME: the name of the OLAP model; for example, TBC Model.
- MODEL_DESCRIPTION: a description of the OLAP model. If you do not enter a description when you save the model, this column is blank.
- MODEL_DATA_SOURCE: the name of the Open Database Connectivity (ODBC) data source on which the OLAP model is based; for example, TBC.
- MODEL_OWNER: the login name of the OLAP model owner; for example, `sa`. The login name is specified in the relational database that contains the OLAP Metadata Catalog.

- MODEL_ACCESS_CODE: a reference number that indicates what level of access users, other than the owner, have to the OLAP model.
  - The number 0 represents no permissions—other users can neither read nor write to the OLAP model.
  - The number 1 represents read access—other users can read the OLAP model but cannot write to it.
  - The number 2 represents read and write access—other users can both read and write to the OLAP model; this is the default.

➤ To view access permissions for all OLAP models in OLAP Metadata Catalog:

1. Issue the following SELECT statement in the OLAP Metadata Catalog database, using the tools for the relational database.

```
SELECT MODEL_NAME,MODEL_OWNER,MODEL_ACCESS_CODE
       FROM OM_INFO
```

2. View the results.

In the following example, TBC Model gives read and write permissions to other users (MODEL_ACCESS_CODE = 2). TBC_Mod_Archive gives read permissions to other users (MODEL_ACCESS_CODE = 1). TBC_Mod_Mine gives neither read nor write permissions to other users (MODEL_ACCESS_CODE = 0).

```
MODEL_NAME           MODEL_OWNER   MODEL_ACCESS_CODE
----------           -----------   -----------------
TBC Model            sa            2
TBC_Mod_Archive      sa            1
TBC_Mod_Mine         sa            0
(3 row(s) affected)
```

Viewing Permissions for Metaoutlines

OLAP Metadata Catalog stores information about the permissions set for OLAP models and metaoutlines. View this information by selecting the appropriate columns from tables in OLAP Metadata Catalog.

The MO_INFO table in OLAP Metadata Catalog contains information about metaoutlines, including the following columns which are relevant to permissions:

- MO_ID: a system-generated reference number for the metaoutline.
- MODEL_ID: a system-generated reference number for the OLAP model on which the metaoutline is based.
- MO_NAME: the name of the metaoutline; for example, TBC Metaoutline.
- MO_DESC: a description of the metaoutline. If you do not enter a description when you save the metaoutline, this column is blank.
- MO_CHANGE_DATE: the date on which changes were last made to the metaoutline.
- MO_OWNER: the login name of the metaoutline owner; for example, `sa`. The login name is specified in the database that contains the associated OLAP Metadata Catalog.
- MO_ACCESS_CODE: a reference number that indicates what level of access users, other than the owner, have to the metaoutline.
  - The number 0 represents no permissions—other users can neither read nor write to the metaoutline.
  - The number 1 represents read access—other users can read the metaoutline but cannot write to it.
  - The number 2 represents read and write access—other users can both read and write to the metaoutline; this setting is the default.

➤ To determine the access permissions for all metaoutlines in OLAP Metadata Catalog:

1. Issue the following SELECT statement in the OLAP Metadata Catalog database, using the tools for the relational database.

   ```
   SELECT MO_NAME,MO_OWNER,MO_ACCESS_CODE
          FROM MO_INFO
   ```

2. View the results.

   In the following example, TBC Metaoutline gives read and write permissions to other users (MO_ACCESS_CODE = 2). TBC_Archive gives read permissions to other users (MO_ACCESS_CODE = 1). TBC_Mine gives neither read nor write permissions to other users (MO_ACCESS_CODE = 0).

   ```
   MO_NAME                MO_OWNER    MO_ACCESS_CODE
   -------                --------    --------------
   TBC Metaoutline        sa          2
   TBC_Archive            sa          1
   TBC_Mine               sa          0
   (3 row(s) affected)
   ```

Chapter

3

Troubleshooting ODBC and Connections

Essbase Integration Services requires several layers of software components to connect to relational databases and load data into Essbase databases. Each of the components must be configured properly to run and communicate with each other. Consequently, diagnosing and correcting connection problems in the system can be challenging.

This chapter assists you in isolating and correcting connection problems that can occur during the operation of Essbase Integration Services. The first topic of this chapter lists common causes for connection problems. The subsequent topics outline a strategy for isolating and correcting connection problems not addressed by the common problems list.

This chapter assumes that you know how to perform basic database administration tasks, such as using a database client utility program to connect to a database server, navigating directories using a command prompt, and editing configuration files.

This chapter contains the following topics:

Checking Common Connection Problems

If you have problems connecting to OLAP Metadata Catalog or to the external data source with Essbase Integration Services Console, review the following list of common problems:

- Are you using the correct user name and password?
  - Does the user name you used have the correct privileges to access both OLAP Metadata Catalog and the data source at the database level?
  - If you are trying to connect to OLAP Metadata Catalog, did you use the same user name and password as the user who created the tables in OLAP Metadata Catalog?

    If you create an OLAP Metadata Catalog when logged in as one user name, you cannot access the tables in OLAP Metadata Catalog using a different user name unless you create an *alias* for the user name (for Sybase and Microsoft SQL Server) or synonyms for the tables (for DB2 and Oracle). For more information, see the *Essbase Integration Services Installation Guide*.
- Are all of the following required components up and running?
  - Essbase Integration Server
  - The relational database management system (RDBMS) listener for OLAP Metadata Catalog and the data source
  - The database servers that manage OLAP Metadata Catalog and the data source databases
- Is the database client software installed and configured on the computer where Essbase Integration Server is installed?
  - Do the database client software utility programs, such as the command-line SQL utility, run? Can you connect to databases with them?
  - Does the user who starts Essbase Integration Server have the correct environment variables set and the necessary access privileges to run the database client software?

  - Are the required environment variables for the database client software, such as the path (and the library path on UNIX), configured on the computer where Essbase Integration Server is installed?

    On UNIX, use `ivteslib` located in the `ISHOME/bin` directory to confirm that the library path contains all files required by the Open Database Connectivity (ODBC) driver.

  - If necessary, is the `bin` directory for each database in the operating system path of the user who is running Essbase Integration Server?
  - Does the ODBC Data Source Name (DSN) configuration identify the name of the relational database management system (RDBMS) client?

    On UNIX, use the `odbcconfig` utility to check the path and environment variable, and to test connectivity to ODBC.

- Are OLAP Metadata Catalog and the data source configured as (ODBC) data sources on the Essbase Integration Server computer? For information on configuring ODBC data sources, see the *Essbase Integration Services Installation Guide*.
- Are you using the ODBC data source names for OLAP Metadata Catalog and the data source databases as configured on the Essbase Integration Server computer? Do not assume that the RDBMS database name is the same as the ODBC data source name for that database.
- Are you experiencing time-out problems when connecting to the external data source?

  Wait and retry the action when the RDBMS is less busy. To avoid the problem, increase the ODBC driver time-out limit using ODBC Administrator on Windows systems. For more information, see the ODBC documentation.

- Did you create tables in OLAP Metadata Catalog? For information on creating tables for the catalog, see the *Essbase Integration Services Installation Guide*.
- Are you having problems related to ODBC while you are creating an Essbase outline? If so, turn on tracing, perform the action again, and analyze the results in the tracing log file. For more information, see “Using ODBC Tracing” on page 3-31.

If you experience problems connecting to the RDBMS, view the log for the data source for information about why the connection failed. For more information, see the documentation for the RDBMS you are using.

If none of the above steps help, review the list of database-specific problems in "Database-Specific Problems" on page 3-4.

Database-Specific Problems

If you have problems connecting to OLAP Metadata Catalog or to the external data source with Essbase Integration Services Console, review the following list of common problems for specific RDBMSs:

- **DB2:** You may be required to bind the database client software utilities to a DB2 server that is managing the databases you want. For more information, see the DB2 installation documentation.
- **Oracle:** In configuring Oracle products to work with ODBC on UNIX systems, you must configure the database names in a `tnsnames.ora` file on the computer that runs Essbase Integration Server. For more information, see the Oracle installation documentation.
- **SQL Server:** If you try to access an SQL Server database with the Microsoft native ODBC driver when you do not have access permission, SQL Server connects you to the default database and does not notify you. Configure the ODBC DSN to use the database that you want to access. For more information, see the SQL Server documentation.
- **Sybase:** To use Essbase Integration Services with Sybase, you must have execute permission for `sp-fkeys` in `Sybsystemprocs`. For more information, see the Sybase documentation.

If none of the previous suggestions help, proceed to "About ODBC and Connection Problems" on page 3-4 for an overview of connection problems, then see "Isolating Connection Problems" on page 3-6.

About ODBC and Connection Problems

There are three layers of software components that Essbase Integration Services Console must pass through to contact an RDBMS: Essbase Integration Server, ODBC, and the database client software. Failure within or between any of these components can cause a lack of communication between the console and the RDBMS.

Diagnosing and correcting connection problems with Essbase Integration Services can prove difficult, because different connection problems often have similar symptoms. In most cases, the errors received in Essbase Integration Services Console are not sufficient to diagnose and resolve the problem, and you must go to the server system to get more information and find a resolution.

The three main categories of problems that break the lines of communication between Essbase Integration Services Console and the RDBMS are:

- **Software component problems:**
  - The components of Essbase Integration Services or the RDBMS are not installed or are installed on the wrong computer.
  - The components do not run because they are not set up properly.
  - The components stopped running or were never started.
  - The components are not compatible with the operating system.
  - The versions of the different components are not compatible with each other.
- **Configuration and communication problems:**
  - The software components are not properly configured to communicate with each other.
  - User names, passwords, and permissions are configured or used inconsistently across the components and computers.
  - The communication interfaces of the software components are incompatible with each other.
  - Changes to components, databases, users, or permissions have put the software components out of sync with each other.
- **Network and security problems:**
  - A physical network path does not exist between the computers that must communicate with each other.
  - One or both computers attempting to communicate do not have the TCP/IP network protocol installed or configured.
  - The computer attempting to contact another computer does not have access to addressing information, such as a domain name server (DNS).
  - Security systems prevent all communication or certain types of communication between the computers that must talk with each other.

Isolating Connection Problems

If the list of problems provided in "Checking Common Connection Problems" on page 3-2 and your own knowledge fail to resolve the connection problem, the recommended strategy for diagnosing the problems is to test the components and connections starting at the RDBMS and backtracking through the connections to Essbase Integration Services Console.

The client software must pass through three layers of components to connect to the RDBMS containing OLAP Metadata Catalog and the data source, as illustrated in Figure 3-1.

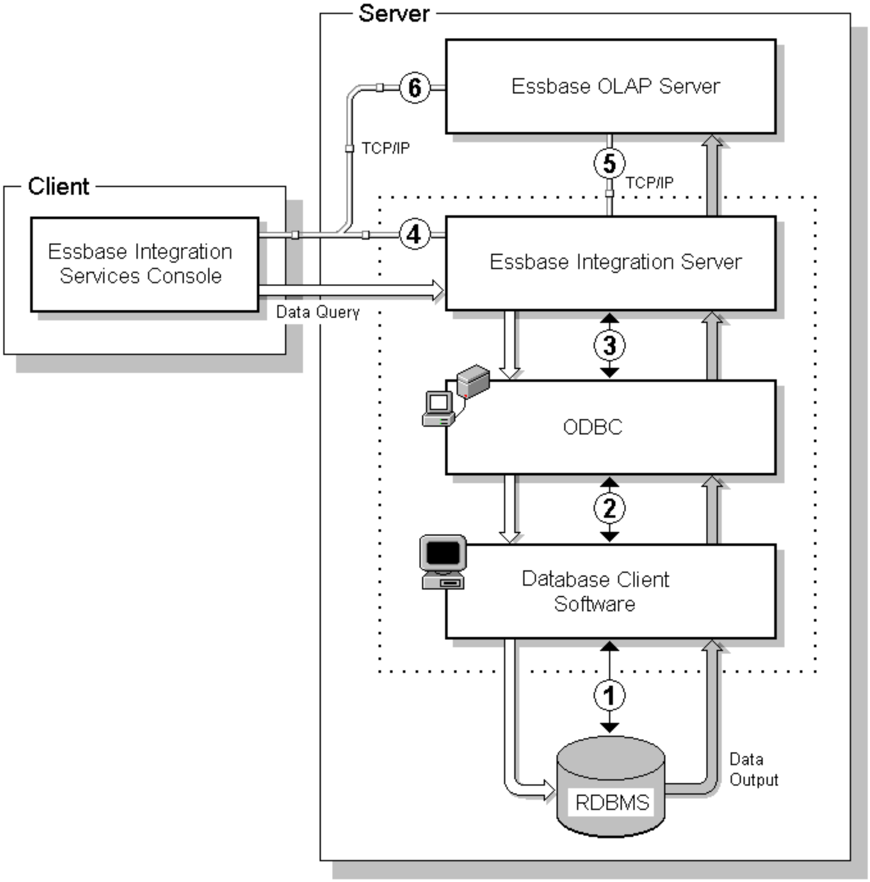


Figure 3-1: Essbase Integration Services Components and Connections

The numbers in Figure 3-1 identify the general diagnostic steps that you should take to isolate connection problems. Follow these steps as indicated, starting with the connection between the database client software and the RDBMS (1) and working backwards through the system to isolate connection problems. After you isolate a problem, refer to the appropriate components topic in "Correcting Connection Problems" on page 3-21.

➤ To isolate connection problems:

1. Attempt to connect to the RDBMS databases using a database client utility program. For more information, see "Testing the Database Client Connection to the RDBMS" on page 3-7.
2. Attempt to connect to the RDBMS databases using ODBC. For more information, see "Testing the ODBC Connection to the RDBMS" on page 3-9.
3. Attempt to connect to the RDBMS databases using Essbase Integration Server. For more information, see "Testing the Essbase Integration Server Connection to the RDBMS" on page 3-15.
4. Attempt to connect to the RDBMS databases using Essbase Integration Services Console. For more information, see "Testing the Essbase Integration Services Console Connection to the RDBMS" on page 3-18.
5. Attempt to connect to Essbase OLAP Server using Essbase Integration Server. For more information, see "Testing the Essbase Integration Server Connection to Essbase OLAP Server" on page 3-19.
6. Attempt to connect to Essbase OLAP Server using Essbase Integration Services Console. For more information, see "Testing the Essbase Integration Server Connection to Essbase OLAP Server" on page 3-19.

Testing the Database Client Connection to the RDBMS

Attempting to connect the database client software to the RDBMS you are using is the first step toward isolating connection problems in Essbase Integration Services. Once you have verified that these components are communicating properly, you have a good base from which to test the remainder of the connection chain.

The most likely cause of a connection failure at this point is that the database client software environment variables are not included in the login script for the user who runs the Essbase Integration Server program.

➤ To test the database client connection to the RDBMS:

1. Log on to the computer running Essbase Integration Server as the user who starts the Essbase Integration Server program (typically, hyperion).

   Note: Logging on as a different user, such as an administrative user, may fail to reveal problems in the configuration.

2. Start a database client utility program with which you can use SQL statements to connect to and query databases, such as a command-line SQL utility.

   If the database client utility does not start, check to make sure the client software is installed and the required environment variables are set. For more information, see "Correcting Database Client Problems" on page 3-25.

3. Connect to the OLAP Metadata Catalog database in the RDBMS using a valid database user account.

   If the connection fails, try to connect as a user who you know has access permission, such as an administrative user. If the administrative connection succeeds, check the other user accounts and permissions in the RDBMS for the OLAP Metadata Catalog database, and carefully check user accounts with connection problems. For more information see, "Correcting RDBMS Problems" on page 3-24.

4. Execute an SQL select statement against a table in the OLAP Metadata Catalog database; for example, if you are connected to the sample OLAP Metadata Catalog, type

   ```
   SELECT * FROM TBC.OV_INFO
   ```

   If the select statement fails, check the permissions for the user name you are using to connect to the RDBMS. For more information, see "Correcting RDBMS Problems" on page 3-24. Also, check that the version of the database client software you are using is compatible with the version of the RDBMS. For more information, see "Correcting Database Client Problems" on page 3-25.

5. After you have successfully completed the preceding steps, try connecting to OLAP Metadata Catalog from Essbase Integration Services Console.

   For more information, see "Testing the Essbase Integration Services Console Connection to the RDBMS" on page 3-18. If the connection fails, proceed to "Testing the ODBC Connection to the RDBMS" on page 3-9.

   If you still cannot complete a connection and select statement with the user who runs Essbase Integration Server, contact technical support for the RDBMS.

6. Repeat steps 3 through 5 for the data source database.

   After you have successfully connected to OLAP Metadata Catalog, you can rule out setup and environment problems on the Essbase Integration Server computer and concentrate on problems specific to the data source database, including required database server names, database names, user names, passwords, and access permissions.

Testing the ODBC Connection to the RDBMS

After you have established that the database client software and RDBMS are communicating properly, the next step in isolating connection problems is to test the ODBC connection to the RDBMS databases.

On Windows NT and Windows 2000, use ODBC Administrator to test connections to the RDBMS databases. In UNIX, either use the `odbcconfig` utility to verify ODBC configuration, manually inspect the ODBC configuration files using `ivtestlib` provided with Essbase Integration Services, or use third-party or RDBMS-provided utilities such as `DemoODBC or tnsping`.

On UNIX systems, the most likely causes for problems in the ODBC component are that the environment variables for ODBC have not been set up or that the `odbc.ini` file has not been configured properly.

ODBC tracing can also be used to track down connection problems. For more information, see "Using ODBC Tracing" on page 3-31.

Testing ODBC on Windows Systems

➤ To test ODBC connections to the RDBMS databases:

1. On the Windows desktop, select **Start > Settings > Control Panel** to open the **Control Panel** window.
2. In the **Control Panel** window, perform one of the following steps:
   - On Windows NT, double-click the **ODBC** icon.
   - On Windows 2000, double-click the **Administrative Tools** icon, and then double-click the **Data Sources (ODBC)** icon.
3. In the **ODBC Data Source Administrator** dialog box, click the **System DSN** tab.
4. In the **System Data Sources** list, select the name of OLAP Metadata Catalog or the data source database, and click **Configure**.

   A data source configuration dialog box is displayed. Depending on the RDBMS and the driver you are using, you may be asked to log on to the database immediately, or you may be asked to review all the settings before testing the connection. In either case, after attempting to log on to the database, observe the resulting message boxes to determine if you connected to the RDBMS successfully.
5. If you cannot connect to the database, check the server name, database name, user name, and password information for accuracy, and make sure that you are using the correct ODBC driver for the RDBMS.

   For more information, see "Correcting ODBC Problems" on page 3-27.
6. After you have successfully completed the preceding steps, try connecting to the RDBMS from Essbase Integration Services Console.

   For more information, see "Testing the Essbase Integration Services Console Connection to the RDBMS" on page 3-18.

   If the connection from Essbase Integration Services Console fails, proceed to "Testing the Essbase Integration Server Connection to the RDBMS" on page 3-15.

Testing ODBC on UNIX Systems

On UNIX systems, you use the `odbcconfig` utility to verify, add, or delete ODBC connections. You can use `odbcconfig` to perform preliminary testing. You must inspect the configuration files for ODBC manually or use an RDBMS utility to test ODBC to perform more thorough testing. These configuration files include the login scripts that set the environment variables for ODBC and the `odbc.ini` file. For more information about ODBC utilities on UNIX systems, see the documentation for the RDBMS or the ODBC driver.

DB2: To use the `odbcconfig` utility with DB2, the environment variable INSTHOME must be set to the DB2 installation directory name, and exported.

Teradata: To use the `odbcconfig` utility with Teradata, the environment variable TDODBC must be set to the Teradata installation directory name, and exported.

➤ To verify and add an ODBC connection using the `odbcconfig` utility:

1. Log in to the computer that runs Essbase Integration Server as the user who starts the Essbase Integration Server software.
2. Start the `odbcconfig` utility.

   Verify that the environment settings displayed by the `odbcconfig` utility are correct.

   To edit and test the environment settings, the available options are: List, Add, Delete, Test, and Exit.
3. To list available DSNs, type **1** and press **Enter**.

   Verify that the DSN exists and, if necessary, proceed to step 4 to add it.
4. To add a DSN, type **2** and press **Enter**.

   At the prompts, type a database type, data source name, and a description. You may also be prompted for additional information depending on the RDBMS platform of the DSN you are adding, such as servername, hostname, or database name.
5. To verify addition of a DSN, type **1** and press **Enter**.

   Verify that the DSN you just added is included in the list.
6. Type **5** and press **Enter** to exit the utility.

You can delete DSNs by typing 3 at the starting `odbcconfig` utility window. Use the List option (1) to verify that the DSN has been deleted.

➤ To test an ODBC connection using the `odbcconfig` utility:

1. Log in to the computer that runs Essbase Integration Server as the user who starts the Essbase Integration Server software.
2. Start the `odbcconfig` utility.

   The utility lists the available options: List, Add, Delete, Test, and Exit.
3. To test the ODBC connection, type **4** and press **Enter**.

   If no errors are returned, the connection is valid. In the event of errors, proceed to step 4 to add corrected DSN information.
4. To add a DSN, type **2** at the starting `odbcconfig` utility window.

   The `odbcconfig` utility does not permit editing of DSN information. You must add new DSN entries and delete any incorrect entries.

   At the prompts, type a data source name, user ID, and password.
5. Type **1** and press **Enter** to list DSNs and verify that the DSN you just added is included in the list.
6. Type **4** and press **Enter** to test the new ODBC connection.

   If no errors are returned, the connection is valid.
7. Repeat the steps in this section until the connection tests return no errors.
8. Type **5** and press **Enter** to exit the utility.

➤ To inspect the login script files:

1. Log on to the computer that runs Essbase Integration Server as the user who starts the Essbase Integration Server software.
2. In the current home directory, find the main login script file (typically, `.profile` for Korn Shell and Bourne Shell users, `.login` for C Shell users) and open it in a text editor.

   Note: The primary login script file may be named differently, depending on the UNIX operating system and the system administrator.

3. Check the main login script file for the inclusion of the Essbase Integration Services script (`is.sh` or `is.csh`).

   If one of these scripts is included in the main login script, make sure that the inclusion of the script file name follows the syntax for executing a script file within another script, and that the script file is specified with a complete directory path. For more information, see “Correcting ODBC Problems” on page 3-27.

   Note: If you make a change to the login scripts, log out and then log back on to reset the environment variables.

4. After completing the preceding steps, test the connection by following the steps listed in “Testing the Essbase Integration Services Console Connection to the RDBMS” on page 3-18.

   If you cannot make a connection, proceed to “Testing the Essbase Integration Server Connection to the RDBMS” on page 3-15.

➤ To inspect the `odbc.ini` file:

1. Log on to the computer that runs Essbase Integration Server as the user who starts the Essbase Integration Server software.

2. To determine the location of the `odbc.ini` file, type

   ```
   echo $ODBCINI
   ```

   If this command does not display a file name and directory location, then you have not included the Essbase Integration Services script in the login script for the user. For more information, see “Correcting Essbase Integration Server Problems” on page 3-28.

3. Move to the directory specified by $ODBCINI and open the `odbc.ini` file with a text editor.

   Ensure that you are using the exact file name and directory that you have set. For example, type **`vi $ODBCINI`** to display the file name and path of the $ODBCINI environment variable.

4. Check that the name of the data sources you are using with Essbase Integration Services are listed in the ODBC Data Sources section; for example

```
[ODBC Data Sources]
TBC_MD=Hyperion Integration Server sample catalog
TBC=Hyperion Integration Server sample database
```

5. For each data source, check that there is a section starting with the name listed in the ODBC Data Sources section, enclosed in brackets; for example

   For Oracle:

```
[myoracle]
Driver=
  /home/hyperion/essintegration/odbc/merant36/lib/ARor815.so
Description=my oracle source
ServerName=mytnsServerName
```

   For DB2:

```
[TBC_MD]
Driver=/home/db2inst1/sqllib/lib/db2.so
Description=DB2 database for sample testing
```

6. Within the section for each data source, verify that an ODBC driver file is specified.

   Make sure that the driver file exists by exiting the text editor and attempting to get a directory listing for that file; for example

```
ls /home/db2inst1/sqllib/lib/db2.so
```

   If the file is listed, use `ivtestlib` to check the actual name and location of the driver, then make the appropriate changes to the `odbc.ini` file.

   Tip: Copy the directory and file name for the driver from the `odbc.ini` file and paste it into an `ls` command.

7. Within the section for each data source, verify that the database name, computer name, and other required information is specified.

   For more information, see "Correcting ODBC Problems" on page 3-27.

Testing the Essbase Integration Server Connection to the RDBMS

After verifying the connections between the ODBC, the database client, and the RDBMS, the next step in isolating a connection problem is to connect to the RDBMS databases from Essbase Integration Server. Test this connection by running Essbase Integration Services Shell on the same computer that runs Essbase Integration Server. For more information on the use and syntax of Essbase Integration Services Shell commands, see Chapter 5, “Using Essbase Integration Services Shell.”

Note: The following procedure uses the sample OLAP Metadata Catalog and sample database. For your tests, substitute the OLAP Metadata Catalog and data source information you are using.

This test uses the LOADMEMBER command to test connections. When the loadmember starts a load, it attempts to make connections in the following order:

1. Data source, using ODBC
2. OLAP Metadata Catalog, using ODBC
3. Essbase server, using TCP/IP

In this test, you set up these connections one at a time and execute a loadmember command each time. By examining the errors each time, you can determine which connections are successful.

➤ To test the Essbase Integration Server connection to the RDBMS:

1. Verify that Essbase Integration Server is running.

   If the server software is not running, start the server. If the server will not start, check that the Essbase Integration Services environment variables are set. For more information, see “Correcting Essbase Integration Server Problems” on page 3-28.

2. On the same computer that is running Essbase Integration Server, start Essbase Integration Services Shell at a command prompt by typing

   ```
   olapicmd
   ```

   If Essbase Integration Services Shell does not start, make sure that the software is installed and that the Essbase Integration Services environment variables are set. For more information, see "Correcting Essbase Integration Server Problems" on page 3-28.

3. At the Essbase Integration Services Shell command prompt, use the login command to connect to the Essbase Integration Server computer; for example, type

   ```
   login cypress
   ```

4. Check the release of Essbase Integration Services by typing

   ```
   version
   ```

   Make sure that the version of the product you are using is compatible with the ODBC drivers and the version of Essbase you are using. For information on component compatibility, see the *Essbase Integration Services Installation Guide*.

5. Set the data source to the sample data source by typing

   ```
   setsource "DSN=TBC;UID=TBC;PWD=password"
   ```

   The DSN parameter requires the ODBC data source name, which may be different from the database name in the RDBMS.

6. Test the connection to the data source database. Attempt to perform a member load by typing

   ```
   loadmember "APP=OLAP_TBC;DBN=Basic;OTL=TBC Metaoutline;"
   ```

7. Check the status of the preceding command by typing

   ```
   status
   ```

   During a member load, Essbase Integration Services Shell connects to the data source first, then to OLAP Metadata Catalog. If the connection to the catalog is unsuccessful, the message "IS Error: Unable to login to Catalog" is displayed. If the connection to the data source fails, the message "IS Error: Unable to login to Datasource" is displayed.

If the connection to the data source failed, verify the ODBC data source name, user name, and password. If the connection continues to fail, see "Correcting Essbase Integration Server Problems" on page 3-28 and "Correcting ODBC Problems" on page 3-27. Repeat steps 4 to 6 of this procedure to retest the connection. When you successfully connect to the data source, continue to the next step.

8. Set OLAP Metadata Catalog to the sample OLAP Metadata Catalog by typing

   ```
   setcatalog "DSN=TBC_MD;UID=TBC;PWD=password"
   ```

   The DSN parameter requires the ODBC data source name, which may be different from the OLAP Metadata Catalog database name in the RDBMS.

9. Test the connection to the OLAP Metadata Catalog database. Attempt to perform a member load by typing

   ```
   loadmember "APP=OLAP_TBC;DBN=Basic;OTL=TBC Metaoutline;"
   ```

10. Check the status of the preceding command by typing

    ```
    status
    ```

    During a member load, Essbase Integration Services Shell connects to OLAP Metadata Catalog and then to the Essbase server. If the connection to OLAP Metadata Catalog is successful, the message "IS Error: Unable to login to Essbase Server" is displayed. If the connection to OLAP Metadata Catalog fails, the message "IS Error: Unable to login to Catalog" is displayed.

    If the connection to OLAP Metadata Catalog failed, verify the ODBC data source name, user name, and password. If the connection continues to fail, see "Correcting Essbase Integration Server Problems" on page 3-28 and "Correcting ODBC Problems" on page 3-27. Repeat steps 7 to 9 of this procedure to retest the connection. When you successfully connect to OLAP Metadata Catalog, continue to the next step.

11. Test the connection from Essbase Integration Services Console to the RDBMS by proceeding to "Testing the Essbase Integration Services Console Connection to the RDBMS" on page 3-18.

Testing the Essbase Integration Services Console Connection to the RDBMS

After you have corrected problems with the components that enable Essbase Integration Services Console to connect to the RDBMS, attempt a connection from the console to an OLAP Metadata Catalog and a data source database in the RDBMS to prove the corrections are successful. To isolate possible computer-to-computer connection problems, run Essbase Integration Services Console on a different computer than the one running Essbase Integration Server.

➤ To test the Essbase Integration Services Console connection to the RDBMS:

1. Verify that Essbase Integration Server is running.

   If the server software is not running, start the server. If the server will not start, check that the Essbase Integration Services environment variables are set. For more information, see "Correcting Essbase Integration Server Problems" on page 3-28.

2. On a computer other than the one running Essbase Integration Server, start Essbase Integration Services Console.

3. Connect to the Essbase Integration Server computer and OLAP Metadata Catalog; for example, cypress and TBC_MD.

   If the connection fails, verify that you have specified the correct ODBC data source name for the OLAP Metadata Catalog database and the correct Essbase Integration Server computer name. Also verify that you specified the correct user name and password for the OLAP Metadata Catalog database on the RDBMS. For more information, see "Correcting Essbase Integration Services Console Problems" on page 3-30.

4. After connecting to OLAP Metadata Catalog, open an OLAP model and connect to the data source for the OLAP model.

   If the connection fails, verify that you have specified the correct ODBC data source name for the data source and the correct Essbase Integration Server computer name. Also verify that you specified the correct user name and password for the data source database on the RDBMS. For more information, see "Correcting Essbase Integration Services Console Problems" on page 3-30.

Testing the Essbase Integration Server Connection to Essbase OLAP Server

When isolating connection problems between Essbase Integration Server and Essbase OLAP Server, use Essbase Integration Services Shell to establish a connection to a data source and OLAP Metadata Catalog, and then attempt to load members into an Essbase database.

➤ To test the Essbase Integration Server connection to an Essbase OLAP Server:

1. Verify that Essbase Integration Server is running.

   If the server software is not running, start the server. If the server will not start, check to ensure that the Essbase Integration Services environment variables are set. For more information, see "Correcting Essbase Integration Server Problems" on page 3-28.

2. Verify that the Essbase OLAP Server is running.

3. At a command prompt, start Essbase Integration Services Shell by typing

   ```
   olapicmd
   ```

   If Essbase Integration Services Shell does not start, make sure that the software is installed and that the Essbase Integration Services environment variables are set. For more information, see "Correcting Essbase Integration Server Problems" on page 3-28.

4. At the Essbase Integration Services Shell command prompt, use the login command to connect to Essbase Integration Server; for example, type

   ```
   login cypress
   ```

5. Set the data source to the sample data source by typing

   ```
   setsource "DSN=TBC;UID=TBC;PWD=password"
   ```

   The DSN parameter requires the ODBC data source name, which may be different from the database name in the RDBMS.

6. Set OLAP Metadata Catalog to the sample OLAP Metadata Catalog by typing

   ```
   setcatalog "DSN=TBC_MD;UID=TBC;PWD=password"
   ```

   The DSN parameter requires the ODBC data source name, which may be different from the OLAP Metadata Catalog database name in the RDBMS.

7. Set the Essbase OLAP Server to which you want to connect by typing

   ```
   settarget "DSN=sequoia;UID=TBC;PWD=password"
   ```

8. Test the connection to the Essbase OLAP Server by attempting to perform a member load by typing

   ```
   loadmember "APP=OLAP_TBC;DBN=Basic;OTL=TBC Metaoutline;"
   ```

9. Check the status of the preceding command by typing

   ```
   status
   ```

   If the connection failed, the message “IS Error: Unable to login to Essbase Server” is displayed. Make sure that the Essbase OLAP Server is running. Verify the name of the metaoutline in Essbase. Also verify the name of the test application and database in Essbase. Repeat steps 7 to 8 of this procedure to retest the connection.

   After you can connect successfully to the Essbase OLAP Server, proceed to “Testing the Essbase Integration Services Console Connection to Essbase OLAP Server” on page 3-20.

Testing the Essbase Integration Services Console Connection to Essbase OLAP Server

When isolating the connection problems between Essbase Integration Services Console and Essbase OLAP Server, connect to OLAP Metadata Catalog and a data source, and then attempt to load members into an Essbase database.

Note: The following procedure uses the sample application, including the sample database, OLAP Metadata Catalog, OLAP model, and metaoutline. For your tests, substitute the application and databases you are using.

➤ To test the Essbase Integration Services Console connection to an Essbase OLAP Server:

1. Verify that Essbase Integration Server is running.
2. Verify that Essbase OLAP Server is running.
3. Start Essbase Integration Services Console.
4. From Essbase Integration Services Console, connect to the Essbase Integration Server computer and OLAP Metadata Catalog; for example, cypress and TBC_MD.

5. After connecting to OLAP Metadata Catalog, open a metaoutline and connect to the data source for the metaoutline; for example, TBC Metaoutline and TBC.
6. Start a member load by selecting **Outline > Member Load**.
7. If prompted, provide the name of the Essbase OLAP Server, user name, and password.

   If the **Essbase Application and Database** dialog box is displayed, the connection is successful. If the connection failed, verify that you have entered the correct computer name for the Essbase OLAP Server and the correct user name and password. For more information, see the Essbase documentation.

   If the Essbase OLAP Server is located on a large network, you may need to provide a more specific server name address, such as “cypress.hyperion.com” rather than simply “cypress.”

Correcting Connection Problems

In correcting connection problems with Essbase Integration Services, consider both component-level and system-level problems. The following topics list possible problems and areas for investigation within the software components that comprise a Essbase Integration Services system.

To correct system-level problems which can occur across software components, review the following issues:

- Are all user names and passwords being used consistently across all the components and computer user accounts? For more information, see “User Name and Password Problems” on page 3-22.
- Do all of the user accounts used in the system have the appropriate permissions and privileges?

  System administrators typically use highly privileged administrative accounts when setting up and testing a system. After the system is set up, users with fewer privileges may be unable to use the system successfully. Check permissions on the following components:

  - User account on the client computer
  - User account on the server computer
  - OLAP Metadata Catalog database

  - Data source database
  - User account for the Essbase OLAP Server

- Are the software components using adequate addressing to identify the computers to which they are connecting?

  Depending on the size and configuration of the computer network, it may be necessary to use a fully-qualified host address to connect to a particular computer. For example, instead of the host name “cypress,” the complete host name “cypress.mydivision.mycompany.com” may be required. Alternatively, you might need to use the IP address number for the server computer; for example, 127.0.0.1.

- Is the user who starts Essbase Integration Server (for example, hyperion) configured to run all the required software?

  The user (for example, hyperion) must have all required environment variables set and permissions to run all of the following software components:

  - Essbase Integration Server
  - ODBC software
  - Database client software
  - RDBMS server (this may not be required)

User Name and Password Problems

Consistent use of user names and passwords is often a problem in installing, configuring, and using Essbase Integration Services, because of the numerous software components and computers that make up the system.

Tip: During installation, configuration, and testing, keep a list of all the user names and passwords you use to create or modify the system, including the applications with which they are associated and their purpose.

In a smaller, single-server implementation of Essbase Integration Services, where Essbase Integration Server, Essbase OLAP Server, and the RDBMS are all run on a single computer, you must manage five user name and password sets and use them consistently:

- User account on the Essbase Integration Services client computer
- User account on the computer that runs all of the server software

- OLAP Metadata Catalog database in the RDBMS
- Data source database in the RDBMS
- Essbase server software

In a fully distributed, three-server implementation, where Essbase Integration Server, Essbase OLAP Server, and the RDBMS are all run on separate computers, you must manage seven user name and password sets and use them consistently:

- User account on the Essbase Integration Services client computer
- User account on the computer that runs Essbase Integration Server
- User account on the computer that runs the RDBMS
- OLAP Metadata Catalog database in the RDBMS
- Data source database in the RDBMS
- User account on the computer that runs Essbase OLAP Server
- Essbase server software

Adding aliases or synonyms to the RDBMS databases adds to the number of user name and password sets you must manage. For more information about database user aliases and synonyms, see the *Essbase Integration Services Installation Guide*.

Tip: During installation and connection testing, use a single set of user names and passwords to avoid confusion. Add user aliases or synonyms after the system is configured and running.

Correcting RDBMS Problems

To correct problems connecting to the RDBMS, investigate the following possible causes:

- Is the RDBMS server computer and software running? Has the server locked up, stopped, or failed?
- Is the RDBMS software installed? Are all of the components required for ODBC connections installed?

  Some RDBMSs may require additional software components beyond the basic server software to enable ODBC connections. The database client software is usually required on the computer that connects to the RDBMS server using ODBC and also may be required on the RDBMS server computer.
- Do the OLAP Metadata Catalog and data source databases to which you are trying to connect exist? Have the tables for OLAP Metadata Catalog been created?

  You must manually create the OLAP Metadata Catalog tables using creation scripts provided with Essbase Integration Services. For more information, see the *Essbase Integration Services Installation Guide*.
- What are the exact names of the OLAP Metadata Catalog and data source databases? Are the names case-sensitive?
- Does the user name that you are using to connect to the RDBMS have adequate permissions to access the database? Does the user have read and write permissions?

  For the OLAP Metadata Catalog database, users need full read and write permissions. For the data source database, users need read-only permissions.

  Sybase: To use Essbase Integration Server on Sybase, you must have execute permission for `sp-fkeys` in `Sybsystemprocs`.
- If the RDBMS server is on a computer by itself, is there a corresponding user account on the computer for the user names you are using to connect to the databases?

  To allow connections to the database server from other computers, you may need to set up user accounts on the RDBMS server computer, separate from the database users you create.

- Does the RDBMS require any addressing beyond the name of the database if you are using the database client software or ODBC to connect?

  Some RDBMSs may have specific naming requirements for connecting to their databases, including different naming conventions for third-party applications (such as Essbase Integration Services). For more information, see the documentation for the RDBMS.

Correcting Database Client Problems

If you have problems connecting to the RDBMS from the database client software, investigate the following possible causes:

- Is the database client software installed? Are all database client components required for the ODBC connections installed?

  To enable ODBC connections, some database client software may require additional software components beyond the basic server software. The database client software usually is required on the computer that uses ODBC to connect to the RDBMS server and may also be required on the RDBMS server computer.

- Is the database client software compatible with the RDBMS?

  Newer or older versions of the database client software may not work with the RDBMS version you are running. For more information, see the RDBMS documentation.

- Do the database client software utilities, such as the command-line SQL interface, start? Do they start when you are logged on as the user who runs Essbase Integration Server?

  When you run the database client software, the user account must have the appropriate file access permissions and must have set the path and other environment variables. Verify that the user who runs Essbase Integration Server (typically, hyperion) is configured to run the database client software.

  On UNIX systems, a script file is usually provided by the RDBMS vendor to set the required environment variables and should be included in the main user login script. If you include one of these files in the main login script, make sure that you specify a complete directory path and that you follow the syntax for executing a script file within a script.

- Are the RDBMS server names, database names, user names, and passwords specified correctly?

  Make sure you are using the correct names, including the exact spelling, uppercase or lowercase letters, and any required RDBMS-specific identification syntax. For more information about correcting user name and password problems, see "Correcting RDBMS Problems" on page 3-24.

- Is the database client software configured correctly?

  File access permissions, path, and other environment variables must be set for the user who runs the database client software. Additional configuration steps also may be required for specific database clients.

  For more information on configuring database client software, see the installation documentation for the database client software.

  DB2: You may be required to bind the database client software utilities to the DB2 databases. For more information, see the DB2 installation documentation.

  Oracle: The database name for Oracle databases can be specified in a `tnsnames.ora` file, which must be configured by the database administrator. For more information, see the Oracle installation documentation.

  Informix: Ensure that the `sqlhosts` environment variable is properly set.

  In the `sqlhosts` file, you must specify a TCP protocol for your operating system as follows:

  - Windows NT, AIX, and HP-UX: onsoctcp
  - Solaris: ontlitcp

  Sybase: Use the dsedit utility (type **`$SYBASE/bin/dsedit`**) to view and edit server entries.

Note: MERANT Version 3.6 ODBC drivers are 32 bit. To connect the database client to Essbase Integration Server using MERANT ODBC drivers, the database client software that you use must be 32 bit.

Correcting ODBC Problems

To correct problems connecting to the RDBMS using ODBC, investigate the following possible causes:

- Is the ODBC software installed? Are all of the required ODBC components installed?

  On Windows systems, make sure the ODBC core components and any required drivers are installed. On UNIX systems, the Essbase Integration Services setup program installs the required ODBC core components if you choose to install the MERANT ODBC drivers.

- Is the ODBC driver compatible with the RDBMS? Is the driver compatible with the operating system?

  Verify that the ODBC driver you are using is supported by Essbase Integration Services by referring to the compatibility matrix provided in the *Essbase Integration Services Installation Guide*.

- Are the RDBMS server names, database names, user names, and passwords specified correctly in the ODBC data source?

  Make sure you are using the correct names, including the exact spelling, uppercase or lowercase letters, and any required RDBMS-specific identification syntax. On Windows systems, configure ODBC data sources by using ODBC Administrator. On UNIX systems, configure ODBC data sources either by using the `odbcconfig` utility or by editing the `odbc.ini` file manually.

  DB2: On Windows systems, the DB2 Client Configuation Assistant shows the ODBC DSN.

  For more information about correcting user name and password problems, see "Correcting RDBMS Problems" on page 3-24.

- On UNIX systems, have you specified the full path and file name of the ODBC driver for the data source in the `odbc.ini` file? Does the ODBC driver actually exist in the specified directory? Can `ivtestlib` load it?

  Try copying the driver path and file name from the `odbc.ini` file and pasting it into an ls command. If the file is not listed when you execute the command, check the accuracy of the driver path and file name in the `odbc.ini` file.

- Are any parameters missing in the ODBC data source configuration?

  On Windows systems, make sure you have filled in all required parameters for the data source in ODBC Administrator. On UNIX systems, the Essbase Integration Services installation provides a sample `odbc.ini` file, which may not include data source parameters required for the RDBMS that you are using. For more information on required RDBMS-specific ODBC driver parameters, see the ODBC documentation provided with Essbase Integration Services in Adobe Portable Document Format (PDF) in the `ISHOME/odbc/merant36/docs` directory. For examples of `odbc.ini` file configurations for the supported RDBMS platforms, see the *Essbase Integration Services Installation Guide*.

- Can other applications connect to the RDBMS by using ODBC?

  If you can identify another application that is successfully using ODBC to connect to databases on the RDBMS, analyze the configuration of the computer from which the connection is made and use it as a model for the computer connections you are trying to fix.

  Note: Most RDBMS vendors provide utilities to test ODBC connections. For information, see the documentation for the RDBMS.

If the preceding checks do not enable you to correct the problems connecting from ODBC to the RDBMS, try using ODBC tracing to isolate the problem. For more information, see "Using ODBC Tracing" on page 3-31.

Correcting Essbase Integration Server Problems

To correct problems connecting to the RDBMS from Essbase Integration Server, investigate the following possible causes:

- Is Essbase Integration Server installed? Are all the required Essbase Integration Server components installed?

  The following components must be installed to have a functional Essbase Integration Server:

  – Essbase Integration Server software.

  – Related base components. (On UNIX, these components are included as part of Essbase Integration Server software.)

  – MERANT ODBC drivers (unless ODBC and drivers are already installed).

- Is Essbase Integration Server running? Is it configured properly?

  Start Essbase Integration Server by opening a command prompt window and typing **`olapisvr`**

  If Essbase Integration Server does not start, review the following possible problems:

  - Are the required environment variables set? Are they set for the correct user name?

    On Windows systems, if you decide not to let the setup program automatically update the environment variables, you must update them manually.

    On UNIX systems, a script file (`is.sh` or `is.csh`) that sets these variables must be included in the login script of the user who starts Essbase Integration Server. For more information, see the *Essbase Integration Services Installation Guide*.

  - What version of Essbase is installed?

    If an incompatible version of Essbase is installed on the same computer as Essbase Integration Server and the `\essbase\bin` directory is in the current path, Essbase Integration Server does not start. For information on compatible versions of Essbase, see the *Essbase Integration Services Installation Guide*.

  - Is port 3388 in use?

    Essbase Integration Server uses port 3388 to communicate with the Essbase Integration Services client software. If this port is being used by another program, the server software does not start.

  - On UNIX, is more than one copy of `olapisvr` running? If so, all servers except one will hang indefinitely. To correct the problem, run the following command to display all programs that are currently running:

    ```
    ps -fe | grep olapisvr
    ```

Many of the problems related to Essbase Integration Server not working properly may be traced to manual configuration steps that you may have missed. For more information about manual configuration steps, see the *Essbase Integration Services Installation Guide*.

Correcting Essbase Integration Services Console Problems

If you have problems connecting to the RDBMS from Essbase Integration Services Console and have ruled out problems with the RDBMS, database client, ODBC, and Essbase Integration Server, the problem likely originates in the Essbase Integration Services Console component.

To correct problems connecting to the RDBMS from the console, investigate the following possible causes:

- Is Essbase Integration Services Console installed? Are all of the required components installed?

  The following components must be installed to have a functional Essbase Integration Services Console client:

  - Essbase Integration Services Console software
  - Related base components

- Is the Essbase Integration Services Console configured properly?

  Make sure the required environment variables are set. If you decide not to have the setup program update the environment variables, you must update them manually. For more information about manual configuration of variables, see the *Essbase Integration Services Installation Guide*.

- Are the Essbase Integration Server computer name, ODBC data source names, user names, and passwords specified correctly?

  For the Essbase Integration Server computer name, verify that you are using the correct spelling and uppercase and lowercase letters for the server name. For OLAP Metadata Catalog and the data source, make sure you are using an ODBC data source name, which may be different from the RDBMS database name.

  Note: The user names and passwords for the OLAP Metadata Catalog database may be different from the data source. Verify that you are using the correct login information for each database.

  For more information about correcting user name and password problems, see "Correcting RDBMS Problems" on page 3-24.

- Does the user account with which you are running Essbase Integration Services Console have permission to connect to the computer running Essbase Integration Server?

  You must have a user account and appropriate access permissions on the server computer running Essbase Integration Server.

- If you have Essbase Integration Server installed on more than one computer, have you specified the correct server name?

  A particular Essbase Integration Server may not be set up to access the same OLAP Metadata Catalogs and data sources used by other servers.

- Are you using adequate addressing to identify the Essbase Integration Server computer?

  Depending on the size and configuration of the computer network, it may be necessary to use a fully-qualified host address to connect to a particular computer. For example, instead of the host name “cypress,” the complete host name “cypress.mydivision.mycompany.com” may be required. Alternatively, you might need to use the IP address number for the server computer; for example, 127.0.0.1.

Using ODBC Tracing

If you do not have access to ODBC testing utilities or third-party applications that use ODBC, tracking down ODBC problems can be difficult. Using the tracing utility provided with ODBC can help identify and resolve connection problems.

Using ODBC Tracing on Windows Systems

On Windows systems, if you cannot connect to the relational data source or OLAP Metadata Catalog, use the tracing utility in ODBC Administrator to learn which ODBC call is failing.

➤ To use the tracing utility in **ODBC Administrator**:

1. Start **ODBC Administrator** on the computer running Essbase Integration Server.

2. Click the **Tracing** tab, specify a log file for the tracing utility, then click **Start Tracing Now**.

   Note: Tracing quickly creates a large log file. Disable tracing when you are finished: On the **Tracing** tab, click **Stop Tracing Now**.

3. Start Essbase Integration Server and Essbase Integration Services Console.
4. From Essbase Integration Services Console, select **Connections > OLAP Metadata Catalog > Connect**, and try to connect to OLAP Metadata Catalog.
5. If you can connect to OLAP Metadata Catalog, open an OLAP model and try to connect to a relational data source.
6. If either connection fails, see the file you specified as the log file in the root directory.

   If necessary, share the information in this log file with Hyperion Technical Support to help solve the connectivity problem.

Using ODBC Tracing on UNIX Systems

On UNIX systems, if you cannot connect to the relational data source or OLAP Metadata Catalog, use the tracing to learn which ODBC call is failing. On UNIX systems, you must edit the `odbc.ini` file to enable and disable ODBC tracing.

➤ To use ODBC tracing on UNIX systems:

1. On the computer running Essbase Integration Server, open the `odbc.ini` file by using a text editor such as vi.
2. Find the section starting with [ODBC], as shown in the following example:

```
[ODBC]
Trace=0
TraceFile=odbctrace.out
TraceDll=/export/home/users/hyperion/is/odbclib/odbctrac.so
InstallDir=/export/home/users/hyperion/is/odbclib
```

3. Set the Trace setting to 1 to enable ODBC tracing.

   Note: Tracing quickly creates a large log file. Disable tracing when you are finished by setting the Trace parameter to 0.

4. Start Essbase Integration Server and Essbase Integration Services Console.
5. From Essbase Integration Services Console, select **Connections > OLAP Metadata Catalog > Connect**, and try to connect to OLAP Metadata Catalog.
6. If you can connect to OLAP Metadata Catalog, open an OLAP model and try to connect to a relational data source.
7. If either connection fails, read the `odbctrace.out` file.

   If necessary, share the information in this log file with Hyperion Technical Support to help solve the connectivity problem.

Chapter 4

Tables in OLAP Metadata Catalog

OLAP Metadata Catalog is a relational database that contains OLAP models, metaoutlines, and the information necessary to retrieve the required data from the relational data source. You can install OLAP Metadata Catalog in any supported data source. See the *Essbase Integration Services Installation Guide* for a complete list of supported data sources.

This chapter describes the tables in OLAP Metadata Catalog. It contains the following topics:

About OLAP Metadata Catalog

OLAP Metadata Catalog contains a collection of related tables. The tables provide a place for you to store OLAP models and metaoutlines. When you load members or data into an Essbase database, Essbase Integration Services retrieves the information stored in OLAP Metadata Catalog to determine:

- What data to retrieve from the relational data source
- What operations to perform on the data
- How to load the data into the Essbase database

CAUTION: It is strongly recommended that you do not alter the table names or data in OLAP Metadata Catalog with tools other than Essbase Integration Services.

Regularly back up the database that contains OLAP Metadata Catalog so that you can restore OLAP Metadata Catalog from backups if it becomes corrupted. Use one of the procedures in Table 4-1, depending on what OLAP Metadata Catalog component you want to restore.

Table 4-1: OLAP Metadata Catalog Restore Procedures

Component To Restore	Procedure
OLAP Metadata Catalog as you installed it, without any OLAP models or metaoutlines that you might have created	Run a script to create the database tables. See the *Essbase Integration Services Installation Guide*.
OLAP Metadata Catalog, including the OLAP models and metaoutlines that you have created	Restore the database that contains OLAP Metadata Catalog from backups. See the documentation for the relational data source.

Tables Relating to the OLAP Model

Table 4-2 describes each table in OLAP Metadata Catalog that stores information about OLAP models. For information about the other tables in OLAP Metadata Catalog, see:

- "Tables Relating to the Metaoutline" on page 4-5
- "Tables Relating to Drill-Through" on page 4-8
- "Miscellaneous Tables" on page 4-9

Table 4-2: OLAP Model Tables

Table Name	Description
MB_FILTER	Contains information about filters in the OLAP model. This information includes the hierarchy filter and system-generated ID numbers for parts of the OLAP model.
MB_TX_RULE	Contains information about hierarchies in the OLAP model. This information includes transformations to perform on the hierarchy and system-generated ID numbers for parts of the OLAP model.
OA_INFO	Contains information about columns in a dimension table or a fact table. This information includes the column name, the column data type, the OLAP model it belongs to, the dimension or fact table it belongs to, the source table and column on which it is based, whether it is hidden in OLAP models and metaoutlines, whether it is used for Drill-Through, whether it is enabled for attributes, and how to create it if it is a user-defined column.
OD_DETAILS	Contains information about dimensions in the OLAP model. This information includes system-generated ID numbers for parts of the OLAP model.

Table 4-2: OLAP Model Tables (Continued)

Table Name	Description
OD_INFO	Contains information about dimensions in the OLAP model. This information includes the dimension name, the OLAP model the dimension belongs to, and whether the dimension is a measures dimension.
OM_INFO	Contains high-level information about an OLAP model: the name of the OLAP model, the description, the name of the data source on which the OLAP model is based, the owner, and access privileges.
OM_PROPERTIES	Contains property information for the OLAP model.
OMB_DETAILS	Contains information about hierarchies in the OLAP model. This information includes the position of each column in the hierarchy, the order in which to build the hierarchy, and system-generated ID numbers for parts of the OLAP model.
OMB_INFO	Contains information about hierarchies in the OLAP model. This information includes the hierarchy name, the OLAP model that it belongs to, and the dimension to which it belongs.
OV_INFO	Contains information about the dimension tables and the fact table in an OLAP model. This information includes the dimension or fact table name, the OLAP model it belongs to, its position in the right frame of the OLAP Model main window, and whether it is a dimension table or a fact table.
OVL_REL_DETAILS	Contains information about logical joins between dimension tables and fact tables. This information includes the name of the first column and the name of the column to which the first column is joined.
OVL_REL_INFO	Contains information about logical joins between dimension tables and fact tables. This information includes the name of the first dimension or fact table and the name of the dimension or fact table to which the first dimension is joined.

Table 4-2: OLAP Model Tables (Continued)

Table Name	Description
OVP_REL_DETAILS	Contains information about physical joins in the source tables. This information includes the name of the first source column and the name of the source column to which the first source column is joined.
OVP_REL_INFO	Contains information about physical joins in the source tables. This information includes the name of the first source table and the name of the source table to which the first source table is joined.
OM_DESCRIPTION	Contains descriptions for a model, its dimensions, its members, and its hierarchies. This information includes the system-generated ID number for each model, dimension, member, or hierarchy that is described.
JOIN_HINTS	Contains information about the database hints defined for a join. This information includes the type of join, the physical table to join, and the text of the join hint.

Tables Relating to the Metaoutline

Table 4-3 describes each table in OLAP Metadata Catalog that stores information about metaoutlines. For information about the other tables in OLAP Metadata Catalog, see:

- “Tables Relating to the OLAP Model” on page 4-3
- “Tables Relating to Drill-Through” on page 4-8
- “Miscellaneous Tables” on page 4-9

Table 4-3: Metaoutline Tables

Table Name	Description
MO_FILTERS	Contains information about metaoutline filters. This information includes the filter name and the actions the filter will perform.
MO_ID	Contains a system-generated reference number that uniquely identifies a metaoutline within an OLAP model.
MO_INFO	Contains high-level information about a metaoutline: the name of the metaoutline, its description, the date it was last changed, the OLAP model on which the metaoutline is based, its owner, its access privileges, and whether the spreadsheet user of Drill-Through reports must enter the Essbase Integration Server name and a user name and password for the external data source.
MO_PROPERTIES	Contains information about metaoutline properties. This information includes the name and value of metaoutline properties, the name of the source table and column that Essbase Integration Services uses to assign the properties, the numeric range that represents a range specified by the user, Global Settings properties, and sort order. Sort order is based on a column in the relational data source, including columns not present in the OLAP model or metaoutline. Sort order can be ascending or descending. Also contains information on association of attributes (by level or by name) to base dimensions.
MOD_ATTRIBUTES	Contains information about attribute dimensions, such as attribute dimension type, alias names, and transformation rules.

Table 4-3: Metaoutline Tables (Continued)

Table Name	Description
MOD_INFO	Contains information about dimensions in a metaoutline. This information includes the dimension name, the OLAP dimension with which it is associated (unless it is a user-defined dimension), the type of dimension (a user-defined dimension, a measures dimension, a standard dimension, or an attribute dimension), and how members are positioned within the dimension.
MS_INFO	Contains information about members in a metaoutline. This information includes the name of the member, the dimension to which it belongs, whether it is a user-defined member or a standard member, the OLAP table with which it is associated, the OLAP model column with which it is associated, the name of its parent, how it is positioned with its siblings, and the database measure with which it is associated (if it is a user-defined member).
MS_PREDICATES	Contains information about members in a metaoutline. This information includes member filters.
MS_TX_RULE	Contains information about members in a metaoutline. This information includes member transformation rules.
MSR_INFO	Contains information about database measures in a metaoutline. This information includes the name of the measure, the OLAP model table with which it is associated, the OLAP model column with which it is associated, and how it is positioned with its siblings.
MSR_PREDICATES	Contains information about database measures in a metaoutline. This information includes measure filters.
MSR_TX_RULE	Contains information about database measures in a metaoutline. This information includes the name and definition of measure transformation rules.
MO_DESCRIPTIONS	Contains descriptions for a metaoutline, its filters, its dimensions, and its members. This information includes the system-generated ID number for each metaoutline, filter, dimension, or member that is described.

Tables Relating to Drill-Through

Drill-through reports are created using the Essbase Integration Services Console OLAP Metaoutline standard user interface (see the *Essbase Integration Services OLAP Metaoutline User's Guide*). These reports enable Essbase Spreadsheet Add-in users to view data in the relational database but which is not stored in the Essbase database.

Table 4-4 describes each table in OLAP Metadata Catalog that stores information about drill-through reports. For information about the other tables in OLAP Metadata Catalog, see:

- "Tables Relating to the OLAP Model" on page 4-3
- "Tables Relating to the Metaoutline" on page 4-5
- "Miscellaneous Tables" on page 4-9

Table 4-4: Drill-Through Tables

Table Name	Description
DTR_ATTRIBUTES	Contains information about drill-through columns. This information includes the order in which Essbase Spreadsheet Add-in sorts and displays drill-through information.
DTR_INFO	Contains information about drill-through columns. This information includes the name of the drill-through column, whether the spreadsheet wizard is displayed, and the time or row query governor specified.
MO_INTERSECTION	Contains information about drill-through columns. This information includes system-generated ID numbers for the OLAP model, the drill-through column, and the intersection level on which the column is defined.

Miscellaneous Tables

Table 4-5 describes tables in OLAP Metadata Catalog that store miscellaneous information about OLAP models and metaoutlines and about Essbase Integration Services. For information about the other tables in OLAP Metadata Catalog, see:

- "Tables Relating to the OLAP Model" on page 4-3
- "Tables Relating to the Metaoutline" on page 4-5
- "Tables Relating to Drill-Through" on page 4-8

Table 4-5: Miscellaneous Tables

Table Name	Description
CB_CONCURRENCY	Contains information about users accessing an OLAP model or metaoutline. This information includes the user name, the user session ID, the OLAP model or metaoutline that the user is accessing, whether the user is accessing an OLAP model or a metaoutline, whether the user has a read or read/write lock, and the date and time the user acquired the lock.
CB_VERSION	Contains information about the Essbase Integration Services software you have installed. This information includes the release number of Essbase Integration Services.

Chapter

5

Using Essbase Integration Services Shell

This chapter describes the commands supported by Essbase Integration Services Shell, the command-line tool provided with Essbase Integration Services. The majority of these commands provide the same functionality available through Essbase Integration Services Console. You use Essbase Integration Services Shell commands and scripts to create Essbase outlines and load Essbase databases.

If you schedule member or data loads using Essbase Integration Services Console, Essbase Integration Services automatically creates an Essbase Integration Services Shell batch file and a corresponding `.cbs` file in the `Batch` directory. You can modify and reuse the `.cbs` file for future member and data loads.

This chapter contains the following topics:

For a list of all available Essbase Integration Services Shell commands, see "Using Essbase Integration Services Shell Commands" on page 5-9.

Starting Essbase Integration Services Shell

Before you start Essbase Integration Services Shell, make sure that the following programs are running. These programs can be on network server computers and do not need to be on your local computer to be available to Essbase Integration Services Shell:

- A relational database management system (RDBMS) with the relational database that contains the OLAP Metadata Catalog where the metadata is stored.
- An RDBMS that contains the relational data source that you want to use to create OLAP models and metaoutlines.
- Essbase Integration Server. For more information, see "Starting Essbase Integration Server" on page 1-2.
- Essbase OLAP Server. For more information, see the *Essbase Integration Services Installation Guide*.

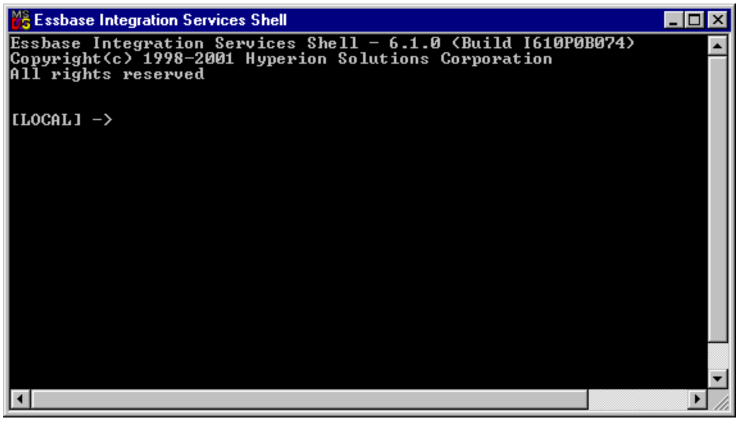
➤ To start Essbase Integration Services Shell, use either of the following methods:

- From the Windows Start menu, select **Programs > Hyperion Solutions > Essbase Integration Services > Essbase Integration Services Shell**.
- Enter the executable name on the command line; for example, type

  ```
  olapicmd
  ```

  Note: If Essbase Integration Services Shell does not start when `olapicmd` is executed from the command line, the operating system path may not be updated correctly. For more information, see the *Essbase Integration Services Installation Guide*.

The Essbase Integration Services Shell window opens, as shown in Figure 5-1:



```
Essbase Integration Services Shell - 6.1.0 (Build I610P0B074)
Copyright(c) 1998-2001 Hyperion Solutions Corporation
All rights reserved


[LOCAL] ->
```

Figure 5-1: Essbase Integration Services Shell Window

In the Essbase Integration Services Shell window, enter commands at the command prompt. For example, type the following commands:

- Type **`login localhost`** to log on to a server running on the same machine.
- Type **`shutdown`** to stop the server.
- Type **`exit`** to stop Essbase Integration Services Shell.
- Type **`help`** to get a list of Essbase Integration Services Shell commands.

About Essbase Integration Services Shell

Essbase Integration Services Shell is a command-line tool that enables you to access Essbase Integration Services to perform operations on an Essbase outline and the data in an Essbase database.

With Essbase Integration Services Shell, you can perform server operations at the command line in either batch or interactive mode.

- Batch mode. To automate routine server maintenance and diagnostic tasks, write a script or batch file and run it from the command line.

  Batch mode is convenient if you use a particular series of commands frequently or if the task requires many commands.

- Interactive mode. To perform tasks interactively, enter OLAP commands at the command line.

  Interactive mode is convenient for short operations that require few commands, such as checking for information on the fly and error checking.

Tip: It can be difficult to enter commands correctly when using the command line. The easiest method is to use the Essbase Integration Services Console OLAP Metaoutline standard user interface to perform a data or member load, and then choose to create a script file. You can then edit the script file as needed. Run all the commands in a script file by redirecting the standard input to use your command file. For example, `olapicmd<`*`mycmd.cbs`*, or see "Running OLAP Command Script Files" on page 5-8.

Using SET and LOAD Commands

SET commands tell Essbase Integration Server which sources are involved in a load operation. LOAD commands tell Essbase Integration Server to execute a load operation.

Note: To perform a successful member and data load, the SET commands must be executed before the LOAD commands.

SET commands provide Essbase Integration Server with the location of the source data, Essbase server, and OLAP Metadata Catalog. The commands can be issued in any order, but you must issue all three SET commands before executing a LOAD command.

Use SET commands as follows:

SETSOURCE	Specifies the relational database from which you want to build an Essbase outline
SETTARGET	Specifies the name of the Essbase server computer on which you want to build an Essbase outline for a Essbase application and database
SETCATALOG	Specifies the OLAP Metadata Catalog that you created to store OLAP models and metaoutlines

The STATUS command (see "STATUS" on page 5-10) does not check or validate the syntax for SET commands.

LOAD commands use the information from the SET commands to load members, data, or both. Use the LOAD commands as follows:

LOADMEMBER	Loads members into an Essbase outline. You must load members before you load data.
LOADDATA	Loads data into an Essbase database. You cannot load data without loading members.
LOADALL	Loads both members and data into an Essbase database.

LOAD commands for small outlines (fewer than 1000 members) take a few seconds to execute. Larger outlines (1000 or more members) take a few minutes to a few hours, depending on the amount of data being loaded, the configuration of the metaoutline, and the computer system capabilities. After executing LOAD commands, wait a few seconds until the command prompt (->) is displayed and then use the STATUS command to check the success or failure of the command execution.

See "Loading Dimensions and Members Using Essbase Integration Services Shell" on page 5-20 for a sample sequence of commands for loading members.

See "Loading Data Using Essbase Integration Services Shell" on page 5-21 for a sample sequence of commands for loading data.

Using Informational Commands

The STATUS and VERSION commands provide processing status and system release version information for you and can be used at any time in the sequence of command execution.

Using Command Syntax and Execution

Essbase Integration Services Shell commands are not case-sensitive, but named data sources, user IDs, and passwords might be case-sensitive, depending on the operating system and the RDBMS you are using. Enter the syntax exactly as shown in this chapter for each command. Keyword parameters can appear in any order.

The commands require semicolons and parameters such as data source name (DSN), user ID (UID), and password (PWD) to separate the command into statements that indicate what kind of information you are providing; for example:

```
SETSOURCE "DSN=database;UID=Smith;PWD=password;"
```

You must separate the statements with a semicolon. The final semicolon is optional. Do not use semicolons as part of data source names, because Essbase Integration Services Shell recognizes semicolons as field separators. It is recommended that you use double quotation marks around the parameters, as shown in the preceding example.

In this *Essbase Integration Services System Administrator's Guide*, all optional syntax parameters are indicated by their enclosure in brackets [ ]; however, do not type the brackets when you enter the commands.

Press Enter to execute the commands.

Note: If you are unsure of a specific syntax, use Essbase Integration Services Console to schedule member or data loads, or create load scripts. An Essbase Integration Services Shell batch file and a corresponding `.cbs` file are created automatically in the *`ISHOME`*`\Batch` directory. You can reference, modify, and reuse the `.cbs` file for future member and data loads. Be sure to check the Save Script Only option instead of Schedule. For more information, see "Recording Member and Data Load Scripts" on page 5-6.

Recording Member and Data Load Scripts

If you are creating scripts to perform loads or other tasks, you may want a script generated by Essbase Integration Services to use as a model or to check your script syntax for Open Database Connectivity (ODBC) data source connection details. Record a load script with Essbase Integration Services by performing a load and recording it to a script file.

➤ To record a load script:

1. Start Essbase Integration Services Console and open a metaoutline.
2. Select **Outline >** and one of the following menu items:
   - Member Load
   - Data Load
   - Member and Data Load
3. In the **Essbase Application and Database** dialog box, type or select the appropriate application and database names.
4. Click **Next** to display the **Schedule Essbase Load** dialog box.
5. Save the load to a script file by clicking **Only Save Load Script** and then click **Save Scripts**.
6. In the **Save Script As** dialog box, type a name for the script that you want to save, and then click **OK**.
7. Click **Finish** to start the load or to complete scheduling of the load.

Running Essbase Integration Services Shell Script and Batch Files

If you use a series of commands frequently or you must enter many commands to complete a task, you can automate the task with a script or batch file. Both are text files.

- A script file (`.cbs` extension) contains Essbase Integration Services Shell commands. You can run a script file from the operating system command line or from within an operating system batch file.
- On Windows NT and Windows 2000, a batch file (`.bat` extension) is an operating system file that calls multiple Essbase Integration Services Shell scripts and is used to run multiple sessions of Essbase Integration Services Shell commands. You can run a batch file on the server from the operating system prompt.
- On UNIX, shell scripts are used in place of batch or script files.

When you run a script file, Essbase Integration Services Shell executes the commands in the order specified in the script until the program reaches the end of the file.

Running OLAP Command Script Files

Enter the following command at the command prompt to run a script file in Essbase Integration Services Shell:

olapicmd -f*scriptFileName* [> *logFileName*]

Replace *scriptFileName* with the name of the script file you are using. Replace *logFileName* with the name of the file where you want the feedback from the script to be recorded.

For example, the following sample script file, `olap_tbc.cbs`, was created in a text editor. This script connects to Essbase from the server console and generates outlines for a sample database. In the following example, the STATUS command (see "STATUS" on page 5-10) is used to check the success or failure of each command. The STATUS command returns the execution status in the Essbase Integration Services Shell window.

```
login localhost
status
settarget "DSN=localhost;UID=hyperion;PWD=password"
status
setcatalog "DSN=TBC_MD;UID=tbc;PWD=password"
status
setsource "DSN=TBC;UID=tbc;PWD=password"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf1;FLT_ID=1;
    CALC_SCRIPT=#DEFAULT#;"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf2;FLT_ID=2;
    OTL_CLEAR=Y;"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf3a;FLT_ID=3;
    OTL_CLEAR=N;CALC_SCRIPT=#DEFAULT#;"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf3ec;FLT_ID=3;
    OTL_CLEAR=N;ESSC_SCRIPT=mytest1"
status
exit
```

To execute the `olap_tbc.cbs` script file, type

```
olapicmd -folap_tbc.cbs
```

The following sample batch file, `olap_tbc.bat`, uses input from a script file named `olap_tbc.cbs` and saves the feedback in a file named `olap_tbc.log`:

```
olapicmd -folap_tbc.cbs > olap_tbc.log
```

Using Essbase Integration Services Shell Commands

The following subtopics describe each command and provide the syntax for entering the command. Each subtopic includes an example command entry for your reference.

The following commands are described in the subtopics:

- "LOGIN" on page 5-10
- "STATUS" on page 5-10
- "VERSION" on page 5-11
- "SETSOURCE" on page 5-11
- "SETTARGET" on page 5-12
- "SETCATALOG" on page 5-12
- "LOADMEMBER" on page 5-13
- "LOADDATA" on page 5-15
- "LOADALL" on page 5-16
- "LOGOUT" on page 5-18
- "SHUTDOWN" on page 5-19
- "EXIT" on page 5-19

Unless otherwise noted, you must be logged on to Essbase Integration Server to execute these commands.

The load commands in this topic are shown with an abbreviated syntax which does not include the syntax required for incremental loads. To see the full syntax for normal or incremental loads, record a script for a load. For more information on recording load scripts, see "Recording Member and Data Load Scripts" on page 5-6.

LOGIN

The LOGIN command connects you to the Essbase Integration Server computer. On a successful connection, the Essbase Integration Services Shell prompt changes to show the name of the Essbase Integration Server computer to which Essbase Integration Services Shell is connected. This command returns an error if you already are connected to the Essbase Integration Server computer.

Syntax

```
LOGIN IShost
```

IShost TCP/IP name or number of the computer where Essbase Integration Server is running

Example

```
LOGIN cypress
```

STATUS

The STATUS command displays the processing status of the most recently issued command. Use it to check the success or failure of all commands. After issuing a command, wait a few seconds until the command prompt (->) is displayed, and then enter the STATUS command.

The STATUS command does not check or validate the syntax for SET commands.

Syntax

```
STATUS
```

Example

Typing STATUS after issuing the LOADMEMBER command returns the following message in the Essbase Integration Services Shell window:

```
LOADMEMBER executed successfully, elapsed time is ’22’ seconds.
```

VERSION

The VERSION command returns the Release number of the Essbase Integration Services software you are running.

Syntax

```
VERSION
```

Example

VERSION returns the release number for Essbase Integration Services software.

SETSOURCE

The SETSOURCE command enables you to identify a source database for Essbase Integration Server to use with a load command.

Syntax

```
SETSOURCE DSN=ODBCdatasource;UID=username;PWD=password[;]
```

ODBCdatasource	The ODBC data source name configured on the computer where Essbase Integration Server runs—case-sensitive
username	The name for logging on to the ODBC data source—case-sensitive
password	The password for logging on to the ODBC data source—case-sensitive

Example

```
SETSOURCE "DSN=TBC;UID=TBC;PWD=Password"
```

SETTARGET

The SETTARGET command enables you to identify a target Essbase server computer for Essbase Integration Server to use with a load command.

Syntax

```
SETTARGET DSN=Essbasedataservername;UID=username;
PWD=password[;]
```

Essbase dataservername	The TCP/IP name or number of the computer where Essbase OLAP Server is running
username	The name for logging on to Essbase OLAP Server
password	The password for logging on to Essbase OLAP Server—case-sensitive

Example

```
SETTARGET "DSN=FIG;UID=TBC;PWD=Password"
```

SETCATALOG

The SETCATALOG command enables you to identify an OLAP Metadata Catalog database for Essbase Integration Server to use with a load command.

Syntax

```
SETCATALOG DSN=ODBCdatasource;UID=username;PWD=password[;]
```

ODBCdatasource	The ODBC data source name configured on the computer where Essbase Integration Server runs—case-sensitive
username	The name for logging on to the ODBC data source—case-sensitive
password	The password for logging on to the ODBC data source—case-sensitive

Example

```
SETCATALOG "DSN=TBC_MD;UID=Fred;TBC=Password"
```

LOADMEMBER

The LOADMEMBER command initiates a member load operation on Essbase Integration Server. The load environment must be set up properly before you invoke the LOADMEMBER command. To set up the load environment, issue the SET commands in any order. For more information, see "Using SET and LOAD Commands" on page 5-4.

To learn the status of the member load invoked by the LOADMEMBER command, wait until you see the command prompt (->), then use the STATUS command.

Syntax

```
LOADMEMBER OTL=Metaoutline;APP=EssbaseApplication;
DBN=EssbaseDatabase;[FLT_NAME=Filtername;]
[FLT_ID=Filter_ID_Number;][ESSC_SCRIPT=EsscmdScriptname;]
[OTL_CLEAR=Delete all members first;][@@USERS=EssbaseUsers;]
[ATTR_UPDATE_LEVEL=AttributeUpdateLevel][;]
```

Metaoutline — The name of the metaoutline—case-sensitive. This metaoutline is used to extract data and create a Essbase outline.

EssbaseApplication — The name of the Essbase application on the target Essbase OLAP Server computer where the Essbase outline is created.

EssbaseDatabase — The name of the Essbase database on the target Essbase OLAP Server computer. Essbase Integration Server applies this name to the Essbase outline.

Filtername/ Filter_ID_ Number — Optional. You can specify a filter name or filter ID to use when loading data. Because the filter name and filter ID are the same, do not specify both. If you do not specify either, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. If the filter name contains a space, enclose it in double quotation marks. For information about creating filters, see the *Essbase Integration Services OLAP Metaoutline User's Guide*.

EsscmdScriptname Optional. The name of an ESSCMD script. The script that you define instructs Essbase to perform specified actions on the Essbase database. These actions might include changing a member tag or loading data from a worksheet into user-defined members after you load the data from the relational data source. The ESSCMD script must be in the `*ISHOME*/esscript` directory.

Delete all members first Optional. Specify Y or N. Enter Y (Yes) for Essbase Integration Server to remove all members from the existing database and then recreate the database by using the members in the metaoutline. This process is slower than creating or updating an Essbase outline without deleting the members; therefore, do not delete all members first unless it is necessary. You should delete all members if, for example, you know that some members have been removed from the OLAP metaoutline and you want to build an Essbase outline containing a smaller set of members. The default is N (No).

EssbaseUsers Optional. Lists all user IDs that can access the Essbase application and database into which you are loading the members. Commas separate user IDs. For example, `@@USERS="admin","ljones","ksmith"`.

AttributeUpdateLevel The level at which attribute dimensions are updated. Values are as follows:

1: Remove all existing attribute dimensions from the Essbase outline and use Global Settings from the OLAP metaoutline.

2: Use Global Settings defined in the Essbase outline to add new attribute dimensions but do not update existing attribute dimensions. With this parameter, no new members will be added to an existing attribute dimension.

3: Use Global Settings from the OLAP metaoutline and perform updates to the existing attribute dimensions. Essbase will report error conditions for any invalid attribute members.

Example

```
LOADMEMBER "OTL=TBC Metaoutline;APP=OLAP_TBC;DBN=Basic;
FLT_NAME=Blue filter"
```

LOADDATA

The LOADDATA command initiates a data load operation on Essbase Integration Server. The load environment must be set up properly before you invoke this command. To set up the load environment, issue the SET commands in any order. For more information, see "Using SET and LOAD Commands" on page 5-4.

LOADDATA takes a few seconds to execute. After executing LOADDATA, use the STATUS command to learn the status of the command execution.

Syntax

```
LOADDATA OTL=Metaoutline;APP=EssbaseApplication;
DBN=EssbaseDatabase;[FLT_NAME=Filtername;]
[FLT_ID=Filter_ID_Number;][CALC_SCRIPT=CalcScriptname;]
[ESSC_SCRIPT=EsscmdScriptname;][@@USERS=EssbaseUsers][;]
```

Metaoutline The name of the metaoutline. Essbase Integration Server uses the specified metaoutline to extract data from the relational data source to create an Essbase outline.

EssbaseApplication The name of the Essbase application on the target Essbase OLAP Server computer where the Essbase outline is created.

EssbaseDatabase The name of the Essbase database on the target Essbase OLAP Server computer. Essbase Integration Server applies this name to the Essbase outline.

Filtername/ Filter_ID_ Number Optional. You can specify a filter name or filter ID to use when loading data. Because filter name and filter ID are the same, do not specify both. If you do not specify either, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. If the filter name contains a space, enclose it in double quotation marks. For information about creating filters, see the *Essbase Integration Services OLAP Metaoutline User's Guide*.

CalcScriptname Optional. The name of the calculation script. The calculation script determines how Essbase calculates the data values in the database. If you do not select a calculation script, Essbase Integration Server does not perform a calculation.

EsscmdScriptname — Optional. The name of an ESSCMD script. The script that you define instructs Essbase to perform specified actions on the Essbase database. These actions might include changing a member tag or loading data from a worksheet into user-defined members after you load the data from the relational data source. The ESSCMD script must be in the `*ISHOME*/esscript` directory.

EssbaseUsers — Optional. Lists all user IDs that can access the Essbase application and database into which you are loading the members. Commas separate user IDs. For example, `@@USERS="admin","ljones","ksmith"`.

Example

```
LOADDATA "OTL=TBC Metaoutline;APP=OLAP_TBC;DBN=Basic;
FLT_NAME=Blue filter"
```

LOADALL

The LOADALL command initiates a load of both members and data on Essbase Integration Server. The load environment must be set up properly before you invoke this command. To set up the load environment, issue the SET commands in any order. For more information, see "Using SET and LOAD Commands" on page 5-4.

LOADALL takes a few seconds to execute. After executing LOADALL, use the STATUS command to learn the status of the command execution.

Note: If you want an ESSCMD to execute between a member and data load, use the LOADMEMBER and LOADDATA commands instead of the LOADALL command.

Syntax

```
LOADALL OTL=Metaoutline;APP=EssbaseApplication;
DBN=EssbaseDatabase;[FLT_NAME=Filtername;]
[FLT_ID=Filter_ID_Number;][CALC_SCRIPT=CalcScriptname;]
[ESSC_SCRIPT=EsscmdScriptname;]
[OTL_CLEAR=Delete all members first;][@@USERS=EssbaseUsers][;]
```

Metaoutline The name of the metaoutline. Essbase Integration Server uses the specified metaoutline to extract data from the relational data source to create an Essbase outline.

EssbaseApplication The name of the Essbase application on the target Essbase OLAP Server computer where the Essbase outline is created.

EssbaseDatabase The name of the Essbase database on the target Essbase OLAP Server computer. Essbase Integration Server applies this name to the Essbase outline.

Filtername/
Filter_ID_ Number Optional. You can specify a filter name or filter ID to use when loading data. Because filter name and filter ID are the same, do not specify both. If you do not specify either, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. If the filter name contains a space, enclose it in double quotation marks. For information about creating filters, see the *Essbase Integration Services OLAP Metaoutline User's Guide*.

CalcScriptname Optional. The name of the calculation script. The calculation script determines how Essbase calculates the data values in the database. If you do not select a calculation script, Essbase Integration Server does not perform a calculation.

EsscmdScriptname Optional. The name of an ESSCMD script. The script that you define instructs Essbase to perform specified actions on the Essbase database. These actions might include changing a member tag or loading data from a worksheet into user-defined members after you load the data from the relational data source. The ESSCMD script must be in the `ISHOME\esscript` directory.

Delete all members first
Optional. Specify Y or N. Enter Y (Yes) for Essbase Integration Services to remove all members from the existing database and then recreate the database by using the members in the metaoutline. This process is slower than creating or updating a Essbase outline without deleting members; therefore, do not delete all members unless it is necessary. You should delete all members if, for example, you know that some members have been removed from the OLAP metaoutline and you want to build the Essbase outline containing a smaller set of members. The default is N (No).

EssbaseUsers
Optional. Lists all user IDs that can access the Essbase application and database into which you are loading members. Commas separate user IDs. For example, `@@USERS="admin","ljones","ksmith"`.

Example

```
LOADALL "OTL=TBC Metaoutline;APP=OLAP_TBC;DBN=Basic;
FLT_NAME=Blue filter"
```

LOGOUT

The LOGOUT command logs you out of Essbase Integration Server. On a successful logout, the Essbase Integration Services Shell prompt changes to LOCAL.

Syntax

```
LOGOUT
```

Example

LOGOUT logs you out of Essbase Integration Server.

Note: If you log out of Essbase Integration Server, you no longer have an active session. To execute another command, you must log on to Essbase Integration Server again.

SHUTDOWN

The SHUTDOWN command shuts down Essbase Integration Server. You must be logged on to Essbase Integration Server before you can use this command.

Syntax

```
SHUTDOWN
```

Example

SHUTDOWN shuts down Essbase Integration Server.

EXIT

The EXIT command exits Essbase Integration Services Shell and closes the Essbase Integration Services Shell window. Before you exit from Essbase Integration Services Shell, Essbase Integration Server automatically executes a LOGOUT command.

Syntax

```
EXIT
```

Example

EXIT logs you out of Essbase Integration Server and closes the Essbase Integration Services Shell window.

Loading Dimensions and Members Using Essbase Integration Services Shell

You can perform a member load by using Essbase Integration Services Shell instead of Essbase Integration Services Console.

You can manually schedule the batch file by using the AT service in Windows NT and Windows 2000 or the cron scheduling daemon on UNIX systems. If you are not sure how to schedule the batch file manually, see the operating system documentation.

➤ To load members and dimensions when using Essbase Integration Services Shell, follow the same basic steps for loading members as you would when using Essbase Integration Services Console:

1. Log on to the Essbase Integration Server computer by using the LOGIN command; for example

```
LOGIN cypress
```

2. Connect to the external data source by using the SETSOURCE command; for example

```
SETSOURCE DSN=TBC;UID=TBC;PWD=password
```

3. Connect to the Essbase server computer by using the SETTARGET command; for example

```
SETTARGET DSN=sequoia;UID=sys;PWD=password
```

4. Connect to OLAP Metadata Catalog by using the SETCATALOG command; for example

```
SETCATALOG DSN=TBC_MD;UID=TBC;PWD=password
```

5. Start the member load by using the LOADMEMBER command; for example

```
LOADMEMBER "OTL=TBC Metaoutline;APP=OLAP_TBC;
DBN=Basic;FLT_NAME=Filter1"
```

Loading Data Using Essbase Integration Services Shell

You can perform a data load by using Essbase Integration Services Shell instead of Essbase Integration Services Console.

You can manually schedule the batch file by using the AT service on Windows NT and Windows 2000 or the cron scheduling daemon on UNIX systems. If you are not sure how to schedule a batch file manually, see the operating system documentation.

➤ To load data by using Essbase Integration Services Shell, follow the same basic steps for loading data as you would using Essbase Integration Services Console:

1. Log on to Essbase Integration Server by using the LOGIN command; for example

   ```
   LOGIN FIG
   ```

2. Connect to the external data source by using the SETSOURCE command; for example

   ```
   SETSOURCE DSN=TBC;UID=TBC;PWD=password
   ```

3. Connect to the OLAP Essbase server by using the SETTARGET command; for example

   ```
   SETTARGET DSN=sequoia;UID=sys;PWD=password
   ```

4. Connect to OLAP Metadata Catalog by using the SETCATALOG command; for example

   ```
   SETCATALOG DSN=TBC_MD;UID=TBC;PWD=password
   ```

5. Start the data load by using the LOADDATA command; for example

   ```
   LOADDATA "OTL=TBC Metaoutline;APP=OLAP_TBC;DBN=Basic;
   FLT_NAME=Filter1"
   ```

Chapter 6

Using the Administration Manager

This chapter describes how to access and use the Administration Manager. Topics in this chapter are:

- “Introducing the Administration Manager” on page 6-1
- “Introducing the Administration Manager” on page 6-1
- “Using the Graphical User Interface” on page 6-2
- “Working with Applications” on page 6-4
- “Working with Databases” on page 6-6
- “Working with User ID’s and Privileges” on page 6-8
- “About Security and Privileges for Databases and Applications” on page 6-11
- “Exporting to Data Warehouse” on page 6-13
- “Exporting to a Text File” on page 6-15
- “Using the Logs” on page 6-15

Introducing the Administration Manager

The Administration Manager is a component of the OLAP Integration Server Desktop from which you can track and manage applications, databases, and users on the current OLAP server.

Opening the Administration Manager

To use the Administration Manager, you must have installed the following components:

- Hyperion Essbase Server
- OLAP Integration Server
- OLAP Integration Server Desktop
- A relational database management system (RDBMS) with the relational data source that contains the OLAP Metadata Catalog

To open the Administration Manager, open the OLAP Integration Server Desktop and select **Tools > Administration Manger** from the menu bar.

Using the Graphical User Interface

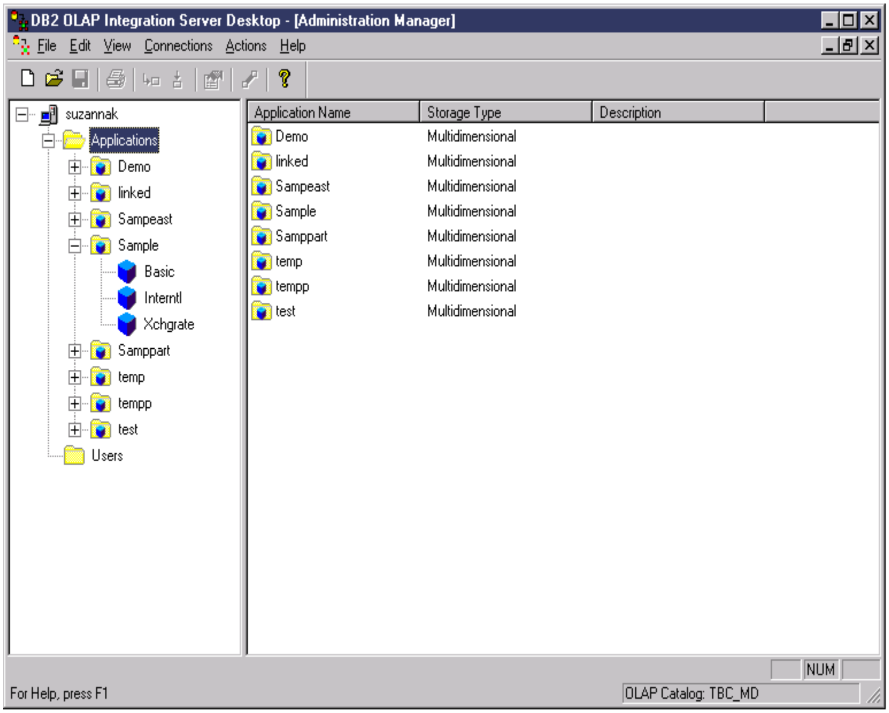
The Administration Manager graphical user interface consists of a right and left pane. The left pane contains a tree view of the databases, applications, and users on the current server. You can expand the tree by clicking the plus (+) sign next to an object to reveal more information about the object. The right pane also displays information about the object that you selected in the left pane.

The root **Server** folder expands to reveal the **Applications** and **Users** folders. The applications are also listed in the right frame when the applications folder is selected. You can sort the applications in the right frame by clicking the column headings. For example, if you want to sort the applications by the type of storage they use, click the **Storage Type** heading.

If you click on the **Users** folder, all the users defined to the applications on the current server are listed in the right frame. You can sort the users by clicking the column headings.

Expand an application folder to reveal database instances within the application. The databases are also listed in the right frame. You can sort the databases in the right frame by clicking the column headings.

Right-click on an application, database, or user object to see what actions are available.



Application Name	Storage Type	Description
Demo	Multidimensional	
linked	Multidimensional	
Sampeast	Multidimensional	
Sample	Multidimensional	
Samppart	Multidimensional	
temp	Multidimensional	
tempp	Multidimensional	
test	Multidimensional	

For Help, press F1

OLAP Catalog: TBC_MD

About the Menu Bar

Objects on the menu bar pertain to either the Model and Metaoutline components of the OLAP Integration Server, or to the Administration Manager.

The **File** and **Connections** options on the menu bar refer to activities such as opening another model or metaoutline, or disconnecting from the current server.

The **Edit, View,** and **Actions** options on the menu bar refer to activities of the databases, applications, and users in the Administration Manager dialog box.

Working with Applications

In the Administration Manager, you can create new applications, start and stop applications, copy or rename applications, and assign user's access to the applications.

Creating New Applications

You can create a new application to load in your metaoutline by using the **New Application** dialog box.

To open the **New Application** dialog box, select New Application from the tree in the left pane.

In the dialog box, you specify a new application name, storage type, and description of the application.

After you have created a new application you can create databases in the application, and assign users who can access the application and databases.

See "Working with Databases" on page 6-6 for information about creating databases.

See "Working with User ID's and Privileges" on page 6-8 for information about assigning users to the applications and databases.

Application Names

Application names can be 1–8 bytes in length and must contain no spaces. If you are using DBCS, only 4 characters are allowed.

You can rename an existing application by using the Rename Application dialog box. To open the dialog box, right-click on the application name and select **Rename** from the list.

Application Storage Types

You can choose either multidimensional or relational storage to store data created by an application. Once you start using a storage type for an application, you cannot change to another storage type. You can use different storage types for different applications on the same server.

Depending on which storage types are enabled by your Essbase Server license, the available storage types might include relational storage or multidimensional storage, or both.

Differences in storage types

With relational storage you can more easily perform any DB2 maintenance or backups of data and perform additional queries and reports than you can on any DB2 database.

With multidimensional storage, data is stored in a proprietary format on the file system on the server. Two files are stored: an index file and a data file, along with the outline. You will experience better performance when you use a multidimensional store.

Starting and Stopping Applications

You can start and stop applications in Administration Manager by right-clicking an application instance in either the right or left pane, then clicking start or stop in the list.

The applications can use a large amount of memory. Depending on the size of your system, you might want to limit the number of applications that you have started at any time. See “Caches” on page 6-6, and “Commit Blocks” on page 6-7 for information about required storage for applications.

Copying Applications

You can copy an existing application by using the **Copy Application** dialog box.

To open the dialog box, right-click on the application name, and select **Copy** from the list.

In the dialog box, type the name of the new application that you are copying to.

Renaming Applications

You can rename an existing application by using the **Rename Application** dialog box.

To open the dialog box, right click on the application name and select **Rename** from the list.

In the dialog box, type the new name that you want for the application.

Application Properties

Use the **Application Properties** dialog box to find what type of storage the application uses, and to list the users who have access to the application.

To open the **Application Properties** dialog box, right-click on an application instance, and select Properties.

The only field you can edit in this dialog box is the **Description** field.

Working with Databases

In the Administration Manager, you can create new databases, copy databases, rename databases, and assign user's access to the databases in an application.

Creating New Databases

Use the **New Databases** dialog box to create a new database in an application.

To open this dialog box, right click an Application instance and select **Add New Database** from the list; or click File > New > Database from the menu bar.

In the dialog box, type a new database name and a brief description of the database. Select values for the cache sizes and commit block size, or accept the defaults. See Caches and "Commit Blocks" on page 6-7.

After you have created a new database, you can assign access to users. See "Working with User ID's and Privileges" on page 6-8 for information about assigning users to databases.

Database Names

Database names can be 1-8 alphanumeric bytes and must contain no spaces. If you are using DBCS, only 4 characters are allowed

Caches

The default cache sizes for a database are:

- Data cache - 3 MB

The valid sizes are from 3 to 4096 MB.

The data cache is an area of virtual storage that contains fully expanded copies of data files available to OLAP.

If you will have many concurrent users, consider doubling the default size.

- Index cache – 10 MB

  The valid sizes are from 1 to 4096 MB.

  The index cache consists of index pages for the current database.

- Data file cache – 32 MB

  The valid sizes are from 8 to 4096 MB.

  The data file cache is an area in memory used by the Storage Manager to compress data blocks before writing them to disk.

  The total cache size for all databases must not exceed the available RAM on your server.

Changes to cache sizes become effective when any of the following events occur:

- You start the database.
- You clear data in the database by issuing a RESETDB command from ESSCMD (not available in DB2 OLAP Starter Kit).
- You clear data in the database by clicking Database > Clear data > All from the menu bar in the Administration Manager or Application Manager. The Application Manager is not available in DB2 OLAP Starter Kit.

Commit Blocks

The default number of commit blocks is 3000 blocks. This is recommended for the Relational Storage Manager (RSM).

If you choose to set the number of commit blocks to 0, blocks will be committed at the end of each transaction. For multidimensional storage, setting commit block size to 0 is the same as allowing committed access; no blocks are committed until all are completed.

Copying Databases

Use the **Copy Database** dialog box to copy a database to another database in the same or different application. To open the dialog box, right click a database instance and select **Copy** from the list. You can also select **Edit > Copy** from the menu bar.

If you are copying the database to a new application, you must first create the new application, then copy the database to the application.

Database Properties

The **Database Properties** dialog box describes the characteristics of the database. It includes the following tabs:

General - this page shows the name of the database, description, and users that have access to the database. None of the fields on this dialog box can be edited except the description field.

Dimensions - this page shows the names of the dimensions in the database, type of dimensions, and information about the numbers of members in the dimensions. None of the fields on this dialog box can be edited.

Data Blocks - this page shows data block parameter and size information. None of the fields on this dialog box can be edited.

Storage Management - use this page to specify the cache sizes and commit block size for your database.

Modifications - this page shows successful data load, calculation, and outline update operations for the current application, including the userid, and start and end times.

Run-Time - this page provides performance data about read and write operations.

To open the **Database Properties** dialog box, right click on a database instance, and select **Properties**.

Working with User ID’s and Privileges

This section tells you how to create or rename users, and grant privileges to access applications and databases. You must have Supervisor authority to create new user ID’s and grant access to databases and applications.

Creating Users

Supervisors can use the **New Users** dialog box to add a new user name and password, and to give users access to applications and databases.

To open the **New User** dialog box, right-click the Users folder in the navigation tree. Select Add from the list.

To add a new user

1. On the **General** tab of the **New User** dialog box, type the user name and password, and confirm the password. Optionally, type a description of the user.
2. Click a radio button to specify a user type of User or Supervisor.

User names can be 1-30 alphanumeric bytes, including spaces.

Passwords can be 1-100 alphanumeric bytes, including spaces. The password must be different than the User Name.

Use the **Access** tab of the **New Users** dialog box to grant access to users for applications and databases. The application and database names that are shown initially are the ones to which the current user (supervisor) has access.

To grant access to an application

1. Click the **Access** tab on the **Add User** dialog box.
2. Click on the application name for which you want to grant access.
3. Click **Change** to open the **Change User Privilege** dialog box.
4. Select the type of access that you want. See “About Security and Privileges for Databases and Applications” on page 6-11.

To grant access to a database

1. On the **Access** tab of the **Add User** dialog box, select the application that contains the databases for which you want to grant access.
2. In the bottom half of the **Access** tab page, select the databases that you want to give access to.
3. Click **Change** to open the **Change User Privilege** dialog box.

4. Select the type of access that you want. See “About Security and Privileges for Databases and Applications” on page 6-11 for descriptions of the access levels.
5. Click **OK**.

Renaming Users

You can rename an existing user by using the **Rename User** dialog box. To open the dialog box, right-click on the user name and select **Rename** from the list.

User Types

The user types are:

- User - Users who have User access initially have no access to applications or databases. Supervisors must assign privileges to Users for access to databases and applications after the userid is created.
- Supervisor – Users who have Supervisor access automatically have Database Designer and Application Designer privileges. Supervisors also have the authority to add, remove, or change database and application access privileges for users and to add other users and supervisors.

Viewing Existing Privileges for a User

Use the **User Properties** dialog box to view the existing privileges for a user.

On the **General** tab, check whether the **User Type** box indicates Supervisor or User status. Supervisor status means that the user has super-user privileges on all applications and databases.

If the user has User status, click the **Access** tab to find what level of access the user has for the applications and databases. The Access tab is not active if the Supervisor radio button is selected.

The User Connections Window

In the Administration Manager, use the User Connections window to view a list of:

- Users who are logged in to the server and the time that they logged in

- Users who are connected to an application or database, and the time that they were connected

To open the User Connections window, click **Security -> Connections.**

Supervisors and application designers can disconnect users from an application or database. However, supervisors and application designers cannot use this window to disconnect themselves. Instead, from the menu bar select **Connections -> OLAP Metadata Catalog -> Disconnect.**

About Security and Privileges for Databases and Applications

This section discusses the various types of user access to databases and applications.

Levels of Privileges for Applications

- None – The user does not have access to the application
- Access Databases – The user has access to the application and can be given access to any or all of the databases in the application.
- Application Designer – The user has authority to do all of the following tasks:

  Modify objects in the application

  Create, modify, and delete databases

  Assign user privileges for the application or databases

  Assign and define filters

  Remove data locks within the application

  Disconnect users from the application

Levels of Privileges for Databases

- None – The user cannot access the database.
- Read – The user has read access to the database.
- Write – The user can read and write to the database.

- Calculate – The user can calculate databases, including databases to which the user does not have read or write access.
- Database Designer – The user has authority to do all of the following tasks:

  Modify objects in the database

  Assign user access privileges

  Assign and define filters

  Remove locks in the database

Relational Database Authorities

The information in this section applies to applications that use the relational storage manager.

The relational storage manager (RSM) in the Essbase server stores data for all applications in one relational database. The RSM always connects to the relational database with the same userid and password. The RSM will use the user ID and password that is stored in the `rsm.cfg.` If no user ID and password exist in the `rsm.cfg` file, the RSM uses the ID that you used to logon to Essbase Server.

Any user ID that is used to connect to the relational database must be given the authority to perform the following tasks in the relational database. See the relational database documentation for instructions on granting authorities and privileges

- Create tables
- Create views
- Create Indexes
- Drop Tables
- Drop Views
- Drop Indexes
- Alter Tables
- Update the data in a table
- Select rows and columns in a table

Global Privilege Settings

Database and application designers can assign a global privilege setting for a database or application. All users who have access to the database or application have, at minimum, the global privilege that is assigned to it.

Exporting to Data Warehouse

The Data Warehouse Center is a component of IBM DB2 Warehouse Manager. You can export a selected database and load script information from DB2 OLAP Server to the Data Warehouse Center and create a step in the Warehouse Center by using the **Export to Data Warehouse Center** dialog box in the Administration Manager.

To open this dialog box, right-click a database instance and select Export to Data Warehouse from the list, or click **Actions > Export** on the menu bar.

Why Export to the Data Warehouse Center?

Initially, when Hyperion Essbase Server loads data into a cube, it gets the data from a source such as DB2. A cube is a set of data and metadata that is stored in a relational database that define a multidimensional database. It refers to the portion of a multidimensional database that is stored in the relational database.

When you want to refresh your data, you can use the Data Warehouse Center to automatically schedule regular updates to your information. Regular updates are useful when you use Essbase Server to generate business reports on a regular basis. The Data Warehouse Center also ensures that the data has been transformed before it is loaded into an OLAP cube. That is, it can access data that might have come from different sources or is in a different format, and ensure the formats are the same.

How the Export Process Works

When the export command is issued from the Administration Manager, OLAP creates a subject area, process, step, and user-defined program in the Data Warehouse Center to build the OLAP cube. When the Data Warehouse Center schedules the step, the user-defined program object executes a batch file. The batch file runs an OLAP Integration Server script that contains commands for building the OLAP cube.

The Role of OLAP Integration Server

The OLAP Integration Server provides the following information when exporting to the Data Warehouse Center:

- Pointers to the data in DB2 that the OLAP Integration Server must access
- The metaoutline (warehouse schema) that defines the data dimensions. The Data Warehouse Center does not contain the data to load into OLAP.
- The full path name to the batch file, which contains the script file
- The OLAP model name
- The OLAP model that was used to create the metaoutline, which defines the dimensions used in the cube. The model is the same as the warehouse schema name in the Data Warehouse Center.
- The table schema name used by the warehouse control database.
- A four-part name consisting of the Essbase Server name, application name, database, outline name (`server.application.database.outline`)
- The script file that contains the pointers to the tables in DB2, the name of the OLAP catalog, the dimensions that are to be accessed, and instructions to OLAP to begin a load process that brings the new data into the OLAP cube. The script is created in the `..\IS\Batch\` directory. When the script file is run by the Data Warehouse Center step, the Essbase Server performs a load operation to update the data in the cube.

The Role of the Data Warehouse Center

The Data Warehouse Center runs the step that is created as a result of the script file. You can schedule the step and promote it so that it runs on a regular basis. The Data Warehouse Center also cleanses the data before it is loaded into the cube.

Sample of Batch and Script files

The following example shows a sample batch file as sent to the Data Warehouse Center from the Essbase Server:

```
C:\IS\Bin\olapicmd < "C:\IS\Batch\TBCModel.cbs" >
"C:\IS\Batch\TBCModel.log"
```

This is a sample HIS script file named C:\IS\Batch\TBCModel.cbs:

```
LOGIN stl01w71
SETSOURCE "DSN=TBC;UID=userid;PWD=password;"
SETTARGET "DSN=olapserv;UID=userid;PWD=password"
SETCATALOG "DSN=TBC;UID=userid;PWD=password;"
LOADALL "APP=app1;DBN=Db1;OTL=TBC Metaoutline;FLT_ID=
1;OTL_CLEAR=N;CALC_SCRIPT=#DEFAULT#;"
STATUS
```

Exporting to a Text File

Exporting data to a text file is a means of backing up data or transferring data to another user. Data is exported to an ASCII file on your server, and is placed in the `ARBORPATH/app` directory.

To open the **Export to a Text File** dialog box, right-click a database instance, and select **Export** from the list; or select Actions > Export from the menu bar.

In the dialog box, you specify the file name that you want to use on the server, and the level of data that you want to export**.** The levels of data that you can export are:

- All data
- Input blocks - blocks that contain data from a data load
- Level 0 sparse data - the smallest, or most granular, branch of each dimension. Only sparse Level 0 dimensions are selected to export to an ASCII file

You can also choose to format the data in columns.

Using the Logs

This section describes the Server Log and the Application Log of the Administration Manager.

The Server Log

The Server Log displays connection, information, and error messages for the server to which you are connected.

To access the Server log, right-click on the server name, then select Log from the list. The Log File Display Options window opens. The Server Log opens after you select a starting date from which to view log files.

The Application Log

The **Application Log** displays event information for the current application.

To access the Application log, right-click on the application name, then select **Log** from the list. The **Log File Display Options** window opens. Specify whether you want to display all log records, or only records starting from a specific date. The Server Log opens after you select a starting date from which to view log files.

Glossary

accounts dimension. A dimension type that makes accounting intelligence available. You can tag only one dimension as accounts; you do not have to have an accounts dimension.

Add Joins mode. In Essbase Integration Services, a state in which you can draw lines to define joins between objects in the OLAP model.

agent log file. A record of actions performed by the agent (server).

aggregate. *See* consolidate.

aggregation level. *See* consolidation level.

alias. An alternative name for a dimension, member, or description.

alias column. In Essbase Integration Services, a column in the data source that contains the aliases for a member level in the metaoutline.

alternate name. *See* alias.

ancestor. A branch member that has members below it. For example, in a dimension that includes years, quarters, and months, the members Qtr2 and 2001 are ancestors of the member April.

application. In Hyperion Essbase, a container for one or more Hyperion Essbase databases and files, such as calculation scripts and data load rules, that are related to the databases.

Architect. *See* Essbase Integration Services Console.

attribute. A classification of a member in a dimension. You can specify an attribute to select and group members with a specified attribute and to perform calculations and application-specific functions. For example, a Product dimension can have several attributes, such as Size and Flavor. A specific member of the Product dimension can have the Size attribute 8 and the Flavor attribute Cola.

base dimension. A standard dimension that is associated with one or more attribute dimensions. To classify a member of a base dimension, you associate it with a member of one or more attribute dimensions that describes the classification, such as a specific flavor. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

block. The primary storage unit within Hyperion Essbase. A block is a multidimensional array representing the cells of all dense dimensions.

branch. A member of a hierarchy that may or may not contain leaf members.

Builder. *See* Essbase Integration Services Console.

calculation script. A text file containing a set of instructions telling Hyperion Essbase how to calculate a database.

catalog. *See* OLAP Metadata Catalog.

cell. A unit of data representing the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet.

column. In relational databases, a vertical part of a table. Also known as a field. A column contains all the values for a specific type of information. Every column has a name and a particular data type. For example, a column may contain the name or employee number for each employee.

Command Interface. *See* Essbase Integration Services Shell.

concatenation. An operation that joins two characters or strings in the order specified, forming one string whose length is equal to the sum of the lengths of the two characters or strings. For example, the strings “New York” and “Library”, when concatenated, become “New York Library”.

condition. In relational databases, a data extraction criterion. For example, you can apply a condition to extract only the data that begins with the letter A.

consolidate. The process of gathering data from dependent entities and aggregating the data to parent entities. Once you enter or load data into child entities, you perform a consolidation to aggregate the data through the organization. As data consolidates, intercompany processing, conversion methods, equity adjustments, and minority ownerships perform calculations on the data.

consolidation level. The top of an aggregation hierarchy or any branch or sub-branch below the top, including the input (leaf) portion of the hierarchy.

data block. *See* block.

data cleansing. The process of making inconsistent data consistent. Examples of inconsistent data are data in which some values are incorrect or not of the correct data type.

data file. A file containing data blocks; Hyperion Essbase generates the data file during a data load and stores it on disk.

data load. The process of populating a Hyperion Essbase database with data. Loading data establishes actual values for the cells defined by the structural outline of the database.

data point. *See* cell.

data type. Defines the kind of data that a column can contain. For example, columns with the Numeric data type contain numbers.

data value. *See* cell.

database. A collection of related information. Each unit (record) of the database is typically organized in a fixed format to make it easy to retrieve selected portions of the data on demand. Each record is made up of one or more data fields, and each data field can hold one piece of data (known as a value).

database administrator. An individual who administers database servers, such as Hyperion Essbase, and who may also design, maintain, and create databases.

database outline. *See* outline.

DateTime transformation. A set of instructions that defines how to change or reformat a relational database DateTime data type to your choice of date format.

DBMS. *See* relational database management system (RDBMS, DBMS).

denormalization. The process of adding redundancy to data in a database, typically by joining tables to form more complete sets of data in the individual tables. This process is performed for the purpose of increasing data retrieval performance. *Contrast with* normalization.

dense dimension. A dimension likely to contain data for every combination of dimension members. For example, a time dimension is typically a dense dimension because it contains all combinations of all members. Contrast with sparse dimension.

detail member. *See* leaf member.

dimension. A data category that is used to organize business data for retrieval and preservation of values. Each dimension usually contains a hierarchy of related members grouped within it. For example, a Year dimension often includes members for each time period, such as quarters and months. Other common business dimensions may be measures, natural accounts, products, and markets.

dimension branch. A collection of dimension tables organized in a hierarchical structure, with one of the dimension tables joined directly to the fact table. A dimension branch defines a single, potential Hyperion Essbase dimension in an Essbase Integration Services metaoutline.

dimension build rules. In Hyperion Essbase, a set of operations similar to data load rules. Instead of loading data, the dimension build rules modify the outline based on data in the external data source file.

dimension table. A container in the OLAP model for one or more relational tables that define a potential Hyperion Essbase dimension. When one dimension table joins to another dimension table, forming a dimension branch, the dimension is composed of the columns of all of the dimension tables in the dimension branch.

dimension type. A property in the OLAP model and in the metaoutline that defines which tables form the Hyperion Essbase accounts, time, and standard (default) dimensions.

duplicate member. The second occurrence of a member name in a data source. Users can determine whether Essbase Integration Server ignores duplicate members or adds them as shared members. *See also* shared member.

Essbase database. A repository of data within Hyperion Essbase that contains a multidimensional data storage array. Each database consists of a defined storage structure (a database outline), data, security definitions, and other associated files, such as calculation scripts or data load rules. *See also* application.

Essbase Integration Server. The server component of the Essbase Integration Services product family. Essbase Integration Server uses the information stored in the OLAP Metadata Catalog to extract the dimension names and member names needed to build a Hyperion Essbase outline from the data source. When the Hyperion Essbase outline is complete, Essbase Integration Server extracts data from the data source, performs the operations specified in the metaoutline, and loads the data into the Hyperion Essbase database.

Essbase Integration Services Console. The client component of the Hyperion Integration Services product family. This graphic interface tool is used to create OLAP models and metaoutlines, and to load data into a Hyperion Essbase database.

Essbase Integration Services Shell. In Essbase Integration Services, a command-line tool that you can use to perform common operations on the Hyperion Essbase outline and the data in the Hyperion Essbase database. For example, you can use the LOADDATA command to load data.

Essbase OLAP Server. A Hyperion server that locates and accesses multidimensional data.

Essbase outline. *See* outline.

fact table. In Essbase Integration Services, a container for one or more relational tables that define the data values for each dimension intersection in the OLAP model. For example, if the OLAP model contains Products, Region, and Year dimensions, the fact table might include data values for the number of units of Product A sold in New York in January.

field. (1) In Hyperion Essbase, a value or item in a data source file that is retrieved from a Hyperion Essbase database. (2) In relational databases, a space allocated for a particular item of information. Fields are the smallest units of information you can access. Most fields have certain characteristics associated with them. For example, some fields are numeric, whereas others are textual. Every field has a name.

file delimiter. One or more characters, such as a comma (,), separating fields in a data source.

filter. (1) In Hyperion Essbase, a method for controlling access to database cells. A filter is the most detailed level of security, allowing you to define the varying access levels that users can have to data. (2) In Essbase Integration Services and in relational databases, a method for controlling which data you retrieve from a relational database. For example, you can choose to retrieve products only in the Product A product family. A filter contains one or more conditions.

foreign key. In relational databases, a column whose data values correspond to the values of a key column in another relational table. *See also* key column and primary key.

formula. In Hyperion Essbase, a combination of operators and functions as well as dimension names, member names, and numeric constants. Formulas are used to calculate relationships between members of a Hyperion Essbase database. @VAR(Actual, Budget) is an example of a formula.

generation. A layer in a hierarchical tree structure that defines member relationships in a Hyperion Essbase database. Hyperion Essbase orders generations incrementally from the dimension (generation 1) down to the leaf members.

generation name. A unique name that describes a generation.

hierarchy. A set of multidimensional relationships in an outline, often created in a tree formation. For example, parents, children, and generations represent a hierarchy.

Hyperion Essbase kernel. A layer of the Hyperion Essbase server that provides the foundation for a variety of functionality, including data loading, calculations, spreadsheet lock&send, partitioning, and restructuring. The Hyperion Essbase kernel reads, caches, and writes data; manages transactions; and enforces transaction semantics to ensure data consistency and data integrity.

Hyperion Integration Server Desktop. *See* Essbase Integration Services Console.

index. (1) In Hyperion Essbase, a method of retrieving data based on sparse dimensions. Also refers to the index files, collectively. (2) In relational databases, pointers that are logically arranged by the values of a key. Indexes optimize access to relational data.

index cache. In Hyperion Essbase, a buffer in memory that holds index pages.

index entry. In Hyperion Essbase, a pointer to an intersection of sparse dimensions. Each index entry points to a block on disk and locates a particular cell within the block by means of an offset.

index file. In Hyperion Essbase, a file used to store data retrieval information. It resides on disk and contains index pages.

index page. In Hyperion Essbase, a subdivision of an index file containing entries that point to data blocks.

input block. A type of data block that has at least one loaded data value.

input data. Any data that is loaded from a data source and is not generated by calculation.

Integration Server. *See* Essbase Integration Server.

integrity constraint. In relational databases, a rule stating that each row should have an entry for each required key column.

Intelligent Help. In Essbase Integration Services, procedural help displayed in a dockable window that accompanies the OLAP Model and OLAP Metaoutline main windows. Intelligent Help provides numbered procedures and links to new automatic-detection options and frequently used functions.

intersection level. In Essbase Spreadsheet Add-in, a Hyperion Essbase member combination that defines a specific value. For example, the member combination Actual, Root Beer, Sales, Jan, East represents the actual January sales value for root beer in the Eastern region.

join. In relational databases, a relationship between two tables based on matching key column values.

join columns. In Essbase Integration Services, two relational table columns that are joined from one table to another.

key column. In relational databases, a column or columns that form a unique identifier for each row. For example, EMPLOYEE_ID might be a key column.

leaf member. A member that has no children.

left frame. (1) In the Essbase Integration Services Console OLAP Metaoutline main window, the area on the left that enables you to view a list of dimensions previously defined in the OLAP model. (2) In the OLAP Model main window, the area on the left that displays a list of the tables and views available in a source relational database.

level. (1) In Hyperion Essbase, a layer of a hierarchical tree structure that defines database member relationships. Hyperion Essbase numbers levels incrementally from the leaf member (level 0) up to the dimension member. (2) In Essbase Integration Services, the distance of a dimension table from the fact table in the OLAP model. For example, if a table is level 2, it is separated from the fact table by two links.

level 0 block. A data block that is created for sparse member combinations when all of the members of the sparse combination are level 0 members.

level 0 member. *See* leaf member.

level name. A unique name that describes a level.

load member. In Essbase Integration Services, a member in a user-defined dimension into which data is loaded. Only user-defined dimensions require load members. For all non user-defined dimensions, Essbase Integration Server knows how to load members and data into the Hyperion Essbase database.

load properties. In Essbase Integration Services, a set of rules that determine what actions the product performs on member level names and data as they are loaded.

logical column. In Essbase Integration Services, a column created by manipulating the data in one or more physical columns. *See also* column. *Contrast with* physical column.

logical table. In relational databases, a table created by manipulating columns from one or more physical tables. The logical table is only a view of the data; the columns remain stored in the original tables and are not physically duplicated in the logical table. *See also* view. *Contrast with* physical table.

mathematical operator. A symbol that defines how data is calculated. A mathematical operator can be any of the standard mathematical or Boolean operators, for example, +, -, *, /, and %. Mathematical operators are used in formulas, and outlines.

MDDB. *See* multidimensional database (MDDB).

measures. Data values that a user wants to track, such as Unit_Price and Discount. By default, measures values map to the accounts dimension in the OLAP model, which maps to the measure dimension in the OLAP metaoutline, which in turn maps to the accounts dimension in the Hyperion Essbase outline.

measures dimension. In Essbase Integration Services, a dimension that, by default, maps to the accounts dimension in the Hyperion Essbase outline.

member. A discrete component within a dimension. For example, a time dimension might include members such as Jan, Feb, and Qtr1.

member combination. In Hyperion Essbase, a list of member names used to specify a set of data at the intersection of two or more dimensions. A member combination is specified by using the cross-dimensional operator -> (a hyphen followed by a right-angle bracket). For example, the actual sales data for the month of January in Sample Basic is Sales->Jan->Actual.

member level. A hierarchical level of detail within a dimension. For example, in a dimension that defines geographic areas by nation, which are then subdivided into provinces, the nation and province categories each represent a member level. A member level corresponds to a level in a Hyperion Essbase outline. The measures dimension contains actual members that are also member levels.

member load. The process of adding new dimensions or members (without data) to a Hyperion Essbase outline. *Contrast with* data load.

metadata. The components of a database outline that describe and hold data. Examples of metadata are dimension names, member names, properties, time periods, and security.

metaoutline. In Essbase Integration Services, a template containing the structure and rules for creating a Hyperion Essbase outline from an OLAP model.

missing data (#MISSING). A marker indicating that data in the labeled location does not exist, contains no meaningful value, or was never entered or loaded. For example, missing data exists when an account contains data for a previous or a future period but not for the current period.

Move mode. In Essbase Integration Services, a state in which you can pick up, move, and drop objects in the OLAP Model main window.

multidimensional. Describes a method of referencing data through three or more dimensions. An individual data value is the intersection of one member from each dimension.

multidimensional database (MDDB). A method of referencing data through three or more dimensions. An individual record is the intersection of a point for a set of dimensions. *Contrast with* relational database.

normalization. The process of grouping and removing redundancy from data so that each entity is in its appropriate place in the database and only in its appropriate place. *Contrast with* denormalization.

numeric transformation. In Essbase Integration Services, a set of instructions that define how to change or reformat a relational database numeric field type. For example, you may choose to divide numeric data by 100.

ODBC. *See* Open Database Connectivity (ODBC).

ODBC data source. Location of the data that you are accessing and the information necessary to access the data using ODBC. For example, an ODBC data source that connects to an SQL Server might require a server name, a database name, a user name, and a password. See also Open Database Connectivity.

OLAP. *See* online analytical processing (OLAP).

OLAP Architect. *See* Essbase Integration Services Console.

OLAP Builder. *See* Essbase Integration Services Console.

OLAP Catalog. *See* OLAP Metadata Catalog.

OLAP Command Interface. *See* Essbase Integration Services Shell.

OLAP Metadata Catalog. In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that you pull from the relational data source. Essbase Integration Server accesses the OLAP Metadata Catalog to generate the SQL statements and the information required to generate a Hyperion Essbase database outline.

OLAP model. In Essbase Integration Services, a logical model (star schema) that you create from tables and columns in a relational database. You can then use the OLAP model to generate the structure of a multidimensional database.

OLTP. *See* online transaction processing (OLTP).

online analytical processing (OLAP). A multidimensional, multi-user, client-server computing environment for users who need to analyze consolidated enterprise data in real time. OLAP systems feature drilling down, data pivoting, complex calculations, trend analyses, and modeling.

online transaction processing (OLTP). OLTP applications are commonly referred to as data capture, data entry, or data collection applications. OLTP applications enable an organization to capture the large amounts of data resulting from its daily activities but provide limited capability for reporting on the data.

Open Database Connectivity (ODBC). Standardized application programming interface (API) technology that allows applications to access multiple third-party databases.

outline. The structure that defines all elements of a database within Hyperion Essbase. For example, an outline contains definitions of dimensions, members, and formulas.

parent. A member that has an aggregated branch below it.

pass-through transformations. In Essbase Integration Services, a feature that allows you to use functions specific to your relational database management system (RDBMS) to extract data values for columns. You can provide a statement that Essbase Integration Services passes through to your RDBMS as a part of the SQL SELECT statement. You provide the statement as a property of a column and the pass-through feature returns a value for the column.

permission. A special privilege that must be assigned to users or groups to enable them to access or modify secure data. Permissions include Read, Read/Write and None.

physical column. A column that is stored in a relational database. *See also* column. *Contrast with* logical column.

physical table. A combination of rows and columns stored in a relational database. *Contrast with* logical table.

pointer. In relational databases, a data element that indicates the location of data in storage.

primary dimension table. A dimension table that joins directly to the fact table. Additional dimension tables may join to the primary dimension table to create a dimension branch.

primary key. In relational databases, a column (key) that uniquely identifies a row. For example, Employee_ID.

query governor. An Essbase Integration Services parameter or Hyperion Essbase OLAP Server configuration setting that you set to control the duration and size of the queries made to the data source.

RDBMS. *See* relational database management system (RDBMS, DBMS).

record. A set of information in a data source. Records are composed of fields, each of which contains one item of information. A set of records constitutes a table. A single record constitutes a row in the table. For example, a table containing personnel information might contain records (rows) that have three fields: a NAME field, an ADDRESS field, and a PHONE_NUMBER field.

recursive table. A source relational table that contains information in one row that is a parent or child of information in another row. For example, in a relational source table containing the columns EMPLOYEE_ID, NAME, and MANAGER_ID, the columns EMPLOYEE_ID and MANAGER_ID are recursive because MANAGER_ID refers back to the EMPLOYEE_ID. Using Essbase Integration Services, you can build a Hyperion Essbase outline hierarchy from a recursive source table.

relational database. A type of database that stores data in the form of related tables. A single database can be spread across several tables, and can be viewed in many different ways. *Contrast with* multidimensional database (MDDB).

relational database management system (RDBMS, DBMS). A system for accessing data in a relational database. This term is often used to describe systems for management of multiple relational databases or tables.

restructure. In Hyperion Essbase, an operation to regenerate or rebuild the database index and, in some cases, the data files.

right frame. In the Essbase Integration Services Console OLAP Metaoutline main window, the area on the right, in which you build a metaoutline; in the OLAP Model main window, the area on the right, in which you build an OLAP model.

schema. In relational databases, a logical model that represents the data and the relationships between the data.

server. *See* Essbase OLAP Server and Essbase Integration Server.

shared member. A member that shares storage space with another member of the same name. The shared member has a property that designates it as shared. The use of shared members prevents duplicate calculation of members that appear more than once in a Hyperion Essbase outline.

sibling. A child member within a dimension, having the same parent as another child member. For example, the members East and West are both children of the Markets dimension and siblings of each other.

sparse dimension. A dimension unlikely to contain data for all combinations of dimension members. For example, product and market dimensions are sparse if not all products are sold in all markets. Contrast with dense dimension.

SQL. *See* Structured Query Language (SQL).

SQL Override. In Essbase Integration Services, a function that enables editing of the standard SQL statements generated by Essbase Integration Server. The edited, user-defined SQL can be selected to improve data load performance when loading data into a Hyperion Essbase database.

staging area. A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMSs.

standard dimension. A dimension that is not an attribute dimension.

star schema. In Essbase Integration Services, a logical model that represents your relational data in a form that mirrors OLAP. A star schema contains a fact table and one or more dimension tables.

storage manager. *See* Hyperion Essbase kernel.

string. A sequence of characters treated as a unit.

Structured Query Language (SQL). A computer language used to access data in relational databases.

system administrator. A person who maintains the hardware, software, disk space distribution, and configurations for running software applications such as Hyperion Essbase.

table. In relational databases, a form of data storage in which data is stored in rows and columns.

transformation rules. In Essbase Integration Services, a set of instructions that define how to change or reformat the member names and data you extract from the source relational database.

UDA. Formerly called user-defined attribute. A UDA is a term associated with members of an outline to describe a particular characteristic of the members. Users can specify UDAs within calculation scripts and reports to return lists of members that have the specified UDA associated with them. UDAs can be applied to dense as well as sparse dimensions.

union. In relational databases, a type of join that combines the results of two SELECT statements. A union is often used to merge lists of values contained in two tables.

user-defined dimension. Dimensions that you explicitly create in Essbase Integration Services, rather than dimensions obtained or built from the relational data source.

user-defined member. Members that you explicitly create in Essbase Integration Services, rather than obtaining and building them from the relational data source.

validation. (1) In Hyperion Essbase, a process of checking a rules file against the outline to make sure the rules file is valid. (2) In Essbase Integration Services, a process of checking the OLAP model and metaoutline.

view. In relational databases, a logical table created by combining columns from one or more tables.

XML Import/Export. In Essbase Integration Services, a function that enables importing data into and exporting data from an OLAP Metadata Catalog in Extended Markup Language (XML) file format. Both OLAP models and metaoutlines can be saved as XML files and imported into other OLAP Metadata Catalogs.

Appendix

A

Essbase Integration Server Error Messages

This appendix details the error messages generated by Essbase Integration Server. Error messages are grouped into the following categories:

- "Member Load Error Messages" on page A-1
- "Data Load Error Messages" on page A-5
- "Drill-Through Report Error Messages" on page A-6
- "Miscellaneous Error Messages" on page A-7

Member Load Error Messages

Table A-1 lists error messages that Essbase Integration Server may generate during a member load. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-1: Essbase Integration Server Member Load Error Messages

Message Number	Message Content
049	Detected metadata join problem (*table.column* < - > *table.column*).
050	Encountered unknown ODBC exception while opening database. Aborting the command.
064	Failed to add database *Essbase database*.
065	Failed to add dimension *dimension*.
066	Failed to add Essbase member *member* (%d).
067	Failed to add Essbase member *member* (child of shared parent).

Table A-1: Essbase Integration Server Member Load Error Messages (Continued)

Message Number	Message Content
068	Failed to add Essbase member *member* (DUPLICATE).
069	Failed to add Essbase member *member* (DUPLICATE, ignored).
070	Failed to add Essbase member *member* (no parent or sibling members found).
071	Failed to add Essbase member *member*, %d.
072	Failed to add Essbase member *member;* the member exists.
073	Failed to add formula *formula*.
074	Failed to add member *member.*
075	Failed to add metaoutline member *member.*
076	Failed to add user attribute *user-defined attribute.*
077	Failed to create a local context.
078	Failed to create a local outline.
079	Failed to create application *Essbase application*.
080	Failed to create dimension *dimension*.
083	Failed to get information for parent of *recursive hierarchy member.*
085	Failed to get metadata for *dimension.member.*
088	Failed to initialize Essbase API.
089	Failed to open outline *Essbase outline*.
093	Failed to restructure *Essbase outline*.
095	Failed to update Essbase server (NULL HEADER)
097	Filter *metaoutline filter* doesn't exist for *metaoutline* metaoutline.
098	Filter expression length exceeded the limit of *maximum bytes* bytes.
115	Incremental update specification *incremental update specification string* is incorrect.
126	Invalid aggregation function *aggregate function* for *table.column*.
129	Invalid esscmd script name *esscmd script.*

Table A-1: Essbase Integration Server Member Load Error Messages (Continued)

Message Number	Message Content
130	Invalid parameter *parameter*.
132	Invalid parameter string.
142	Metadata error: *property ID* is not a valid property for *member* or *dimension*.
143	Metaoutline validation error. *Recursive hierarchy member*, a recursive member cannot have aggregation filter.
144	Metaoutline validation error. Filter = *member filter*, is not associated with any member.
188	Unknown exception encountered while opening dimension *dimension*.
189	Unknown exception encountered while opening member *member*.
190	Unknown exception encountered while opening metaoutline *metaoutline*.
341	Failed to Login to Essbase Server.
342	Failed to Get Essbase Settings.
344	Failed to unload database.
345	Failed to delete database.
346	Failed to close outline.
347	Failed to read outline.
349	Failed to unlock Database *database*.
350	Failed to save outline.
351	Unable to process state for *Essbase command*.
352	Failed to get access permissions for application.
353	Failed to set access permissions for application.
354	Failed to get access permissions for database.
355	Failed to set access permissions for database.
356	No member found.

Table A-1: Essbase Integration Server Member Load Error Messages (Continued)

Message Number	Message Content
357	Essbase Error: *error definition*.
358	Failed to get user attributes.
360	Failed to add member to outline.
361	Failed to calculate the data.
362	Failed to get parent.
363	Failed to get member.
364	Failed to move member.
365	Failed to get shared member.
366	Failed to get member information.
367	Failed to sort outline.
377	IS Error: Unable to delete outline.
378	IS Error: SQL Generation for member load failed.
379	IS Error: Database close operation failed after load.
380	IS Error: Catalog close operation failed after load.
381	IS Error: Essbase close operation failed after load.
382	IS Error: No OLAP model name specified.
383	IS Error: Failed to open OLAP model *OLAP model*.
384	IS Error: Failed to close OLAP Metadata Catalog *OLAP Metadata Catalog*.
385	Failed to create virtual model.
386	Member load completed with errors.
387	Member load terminated due to error.

Data Load Error Messages

Table A-2 lists the error messages that Essbase Integration Server may generate during a data load. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-2: Essbase Integration Server Data Load Error Messages

Message Number	Message Content
034	Cell load exceptions encountered.
047	Data load exceptions encountered. *? , amount*.
063	Failed to activate *Essbase_application.Essbase.database*.
082	Failed to execute esscmd script *esscmd script*.
388	IS Error: Failed to get dataload SQL.
389	IS Error: Failed to get dimbuild SQL.
390	IS Error: No. of dataload SQLs not equal to no. of hierarchies.

Drill-Through Report Error Messages

Drill-Through reports are created using the Essbase Integration Services Console OLAP Metaoutline standard user interface and are viewed using Essbase Spreadsheet Add-in for Excel or Lotus 1-2-3.

Table A-3 lists the error messages that Essbase Integration Server may generate to Essbase users viewing Drill-Through reports. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-3: Essbase Integration Server Drill-Through Report Error Messages

Message Number	Message Content
081	Failed to enable DTS Member *member*.
087	Failed to get the universal member handle.
125	Intersection element *member* does not exist in *table.column*.
131	Invalid parameter count for the drill-through request. Aborting...
153	Received NULL pointer in DTAttributes( ) for Report = *Drill-Through report*.
154	Received NULL pointer in GetDTData ( ) for Report = *Drill-Through report*.
155	Received NULL pointer in GetDTDomain ( ).
156	Received NULL pointer in GetDTReport ( ).
157	Received NULL pointer in SetDTReport ( ).
183	Too many Drill-Through reports defined (exceeded Essbase metadata size limit). Drill-Through is disabled for this outline.
186	Unknown exception in GetDTReport ( ).
359	Essbase Error: Invalid Drill-Through Metadata.
369	IS Error: Data load completed with errors.
370	IS Error: Data load terminated due to errors.
371	IS Error: Unable to log in to data source.

Table A-3: Essbase Integration Server Drill-Through Report Error Messages (Continued)

Message Number	Message Content
372	IS Error: Unable to log in to OLAP Metadata Catalog.
373	IS Error: Unable to log in to Essbase Server.
374	IS Error: Unable to read Metaoutline information.
375	IS Error: Data calculation failed.
376	IS Error: Esscmd script execution failed.

Miscellaneous Error Messages

Table A-4 lists the miscellaneous error messages that Essbase Integration Server may generate during OLAP model and metaoutline creation and during data and member loads. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-4: Essbase Integration Server Miscellaneous Error Messages

Message Number	Message Contents
004	*Metaoutline* failed validation. Database measures not specified.
007	*Add job* failed.
017	*Syntax* Syntax error at *character location* in the filter expression *filter*.
018	*Member* cannot have aggregation filter. Only leaf members can have aggregation filters.
054	Error message unavailable for this error.
057	Essbase server on *Essbase server computer* is not enabled with Essbase Integration Services option. Please inform your system administrator.
084	Failed to get job info (OS error *Job-ID*).

Table A-4: Essbase Integration Server Miscellaneous Error Messages (Continued)

Message Number	Message Contents
086	Failed to get the ODBC message.
090	Failed to remove *temporary batch file* (OS error *operating system error*).
091	Failed to remove job (OS error *operating system error*).
092	Failed to remove old job (OS error *operating system error*).
094	Failed to schedule job (OS error *operating system error*).
096	Fatal error.
124	Internal system error. Please contact Technical Support with the error number 1999999.
127	Invalid command line option *OLAP Integration Server switch*.
128	Invalid datatype specification *data type*.
133	Invalid session identifier; please log in to Essbase Integration Services again.
159	Replace Failed *member transformation*.
185	Trying to access invalid memory. Contact Technical Support.
187	Unknown ODBC exception encountered while closing database.
206	*Syntax* Syntax error at *character location* in the filter expression *filter*.
207	Essbase Integration Services is already running on this machine.
208	Failed to release locks.
333	ODBC Error: Encountered unknown ODBC exception while opening database.
334	ODBC Error: Encountered unknown ODBC exception while closing database.
336	ODBC Error: Failed to get ODBC Error message for error number.
338	IS Error: Error during parsing Load Parameters.

Table A-4: Essbase Integration Server Miscellaneous Error Messages (Continued)

Message Number	**Message Contents**
343	Essbase Error: Essbase Server on *Essbase server computer* is not enabled with Essbase Integration Services option.
344	Failed to unload database.
348	Failed to get information for Database *database*.
391	IS Error: Unable to get tables for pattern name *SQL regular expression pattern* from data source.
392	IS Error: Unable to get columns for table *relational table* from data source.
393	IS Error: Unable to extract foreign keys for tables *relational table, relational table* from data source.
394	IS Error: Failed to bind columns.
395	IS Error: Failed to fetch next row.
396	IS Error: Failed to get number of columns.
397	IS Error: Failed to get column attributes.
398	IS Error: Failed to get statement handle.
399	IS Error: Failed to release statement handle.
400	IS Error: Failed to get number of rows.
401	IS Error: Failed to connect to Essbase Integration Services.

Index

Index