Essbase® Integration Services

Release 7.1

System Administrator's Guide

Hyperion®

Hyperion Solutions Corporation

The information contained in this manual is subject to change without notice. Hyperion shall not be liable for errors contained herein or consequential damages in connection with the furnishing, performance, or use of this material.

Hyperion Solutions Corporation
1344 Crossman Avenue
Sunnyvale, California 94089

Printed in the U.S.A.

Contents

Preface

Purpose

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

Audience

This guide is for Integration Services database administrators who are responsible for implementing, configuring, and managing 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 or a data warehouse)
- A fundamental understanding of database administration, such as use of database tables, columns and records; user management, operation of database client utilities; and execution of SQL commands
- 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

Document Structure

This document contains the following information:

- Chapter 1, “Using Essbase Integration Server,” describes how to start and stop Integration 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 Integration Services Shell.
- Appendix A, “Return Codes and Error Messages,” details the error messages generated by Integration Server.
- Appendix B, “Integration Services Limits and Guidelines,” contains limits and guidelines for Integration Services.
- Glossary, a list of key Integration Services terms and their definitions.
- Index, a list of Integration Services terms and their page numbers. In the online version, select an index entry to view that page.

Where to Find Hyperion Documentation

All Integration Services documentation is accessible from the following locations:

- *Essbase Integration Services Information Map* provides access to all Integration Services documentation in PDF, Windows Help, and HTML formats. The Information Map is located at

  ```
  ISHOME/essintegration/docs/eis_info_map.htm
  ```

- Online help is accessible from Integration Services Console. Start the product and click the Help button on dialog boxes or use the Help menu command. Online help includes procedural and conceptual information as well as dialog box help and a glossary.

- Intelligent Help is displayed in a dockable, relocatable window within Integration Services Console and provides procedural information along with links to automatic-detection options and frequently used functions.
- The Hyperion Solutions Web site is located at `http://www.hyperion.com`.
- The Hyperion Download Center can be accessed from `http://hyperion.subscribenet.com`. or from `http://www.hyperion.com`

➤ To access documentation through the Hyperion Solutions Web site:

1. Log on to `http://www.hyperion.com`.
2. Select the **Support** link and type your username and password to log on.

   Note: New users must register to receive a username and password.

3. Click the **Hyperion Download Center** link and follow the on-screen instructions.

➤ To access documentation from the Hyperion Download Center:

1. Log on to `http://hyperion.subscribenet.com`.
2. In the **Login ID** and **Password** text boxes, enter your assigned login ID name and password.
3. In the **Language** list box, select the appropriate language and click **Login**.
4. If you are a member on multiple Hyperion Download Center accounts, select the account that you want to use for the current session.
5. Perform one of the following actions:
   - To access documentation online, from the **Product List**, select the appropriate product and follow the on-screen instructions.
   - To order printed documentation, from the **Information** section in the left frame, select **Order Printed Documentation**, then follow the on-screen instructions.

➤ To order printed documentation if you do not have access to the Hyperion Download Center:

- In the United States, call Hyperion Solutions Customer Support at 877-901-4975.
- From outside the United States, including Canada, call Hyperion Solutions Customer Support at 203-703-3600. Clients who are not serviced by support from North America should call their local support centers.

Conventions

The following table 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 in procedural steps highlights major interface elements.
CAPITAL LETTERS	Capital letters denote commands and various IDs. (Example: LOADALL command)
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.
`Example text`	Courier font indicates that the material shown is a code or syntax example.
`Courier italics`	Courier italic text indicates a variable field in command syntax. Substitute a value in place of the variable shown in Courier italics.

Table i: Conventions Used in This Document (Continued)

Item	Meaning
ARBORPATH *ISHOME*	When you see the environment variable *ARBORPATH* in italics, substitute the value of ARBORPATH from your site. When you see the environment variable *ISHOME* in italics, substitute the value of ISHOME from your site.
Italics	Italics in a product-related term in the body of a book indicates that the term is also included in the glossary of the book.
n, x	Italic *n* stands for a variable number; italic *x* can stand for a variable number or an alphabet. These variables are sometimes found in formulas.
Ellipses (...)	Ellipsis points indicate that text is omitted from an example.
Mouse orientation	This document provides examples and procedures using a right-handed mouse. If you use a left-handed mouse, adjust the procedures accordingly.
Menu options	Options in menus are shown in the following format. Substitute the appropriate option names in the placeholders, as indicated. ***Menu name*** > ***Menu command*** > ***Extended menu command*** For example: **1.** Select **File** > **Desktop** > **Accounts**.

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 providing documentation and online help, Hyperion offers the following product information and support. For details on education, consulting, or support options, click the Services link on the Hyperion Web site at `http://www.hyperion.com`.

Education Services

Hyperion offers instructor-led training, custom training, and eTraining covering all Hyperion applications and technologies. Training is geared to administrators, end users, and information systems (IS) professionals.

Consulting Services

Experienced Hyperion consultants and partners implement software solutions tailored to clients' particular reporting, analysis, modeling, and planning requirements. Hyperion also offers specialized consulting packages, technical assessments, and integration solutions.

Technical Support

Hyperion provides enhanced electronic-based and telephone support to clients to resolve product issues quickly and accurately. This support is available for all Hyperion products at no additional cost to clients with current maintenance agreements.

Documentation Feedback

Hyperion strives to provide complete and accurate documentation. We value your opinions on this documentation and want to hear from you. Send us your comments by clicking the link for the Documentation Survey, which is located on the Information Map for your product.

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:

About Integration Server

Integration Server is multithreaded server software that serves as the bridge between the data source, OLAP Metadata Catalog, and Essbase Analytic Server. Essbase Integration Services 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 Analytic Services database

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

Starting Integration Server

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

- If you are accessing a relational data source, make sure that the following components are running:
  - The relational database management system (RDBMS) with the relational data source that contains the OLAP Metadata Catalog where the metadata is stored
  - One or more RDBMSs that contain the relational data sources you want to use to create or modify an OLAP model or metaoutline
- If you are accessing a data warehouse, make sure that the following components are running:
  - The data warehouse
  - The SAP BW server if you are accessing an SAP Business Information Warehouse (BW)
  - Essbase Integration Services for SAP BW

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

You can start Integration Server by using one of the following methods:

- From a DOS command prompt
- From a UNIX command prompt
- From the Windows Desktop Start menu
- As a Windows service

Note: To start Integration Server as a Windows service, you must first enable the Windows service for Integration Server. See the *Essbase Integration Services Installation Guide* for more information.

➤ To start Integration Server from the DOS command line:

1. At the command line, type the `startup.bat` command; for example, type

   ```
   startup.bat
   ```

   The `startup.bat` command ensures that the environment is set correctly and that the proper PATH and CLASSPATH settings are used when starting Integration Server.

2. If you want to change any default startup switches, add the switches to the `eis.cfg` file or the `startup.bat` file.

   If you want to change any default configuration parameters, add the parameters to the `eis.cfg` file.

See "Storing Startup Switch Information in the Configuration File" on page 32 and "Storing Startup Switch Information in startup.bat (Windows Only)" on page 34 for information on adding startup switches and configuration parameters to these files.

➤ To start Integration Server from the UNIX command line:

1. Type the executable name on the command line, for example, type

   ```
   olapisvr
   ```

   Note: If 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*.

2. If you want to change any default startup switches, type the command syntax after the `olapisvr` command or add the switches to the `eis.cfg` file.

   See "Storing Startup Switch Information in the Configuration File" on page 32 and "Storing Startup Switch Information in startup.bat (Windows Only)" on page 34 for information on adding startup switches to these files.

   The values for startup switches that you type at the command line override any startup switch values you have stored in the `eis.cfg` file.

   When entering parameters for startup switches at the command line, enclose path names that contain spaces in quotation marks (`""`). The following example illustrates the syntax for passing the -E and -M options:

   ```
   olapisvr -E"c:\my logs\eislog" -M"c:\ess 65\bin\essbase.mdb"
   ```

➤ To start Integration Server from the Windows Desktop Start menu:

1. On the Windows Desktop, click **Start**.
2. Select **Programs** > **Hyperion Solutions** > **Essbase Integration Services** > **Essbase Integration Server** to display the Integration Server window.

   If you have stored startup switch settings in `startup.bat` or `eis.cfg`, the settings in `startup.bat` override the settings in `eis.cfg` when starting Integration Server from the Windows Desktop **Start** menu.

➤ To start Integration Server as a Windows service:

1. Access the **Services** dialog box.

   - On Windows 2000, select **Start** > **Settings** > **Control Panel**, double-click **Administrative Tools**, and then double-click **Services**.
   - On Windows XP, select **Start** > **Control Panel**. In the **Control Panel**, select **Performance & Maintenance**, and then **Administrative Tools**. In the **Administrative Tools** window, double-click **Services**.

   The **Services** dialog box is displayed with **Essbase Integration Server** in the list of services.

2. Select, but do not double-click, **Essbase Integration Server**.
3. If you want to change any default startup switches or configuration parameters, add the switches to the `eis.cfg` file or the `startup.bat` file.
4. Start the service for Integration Server.

   - On Windows 2000, select **Action** > **Start**.
   - On Windows XP, click **Start**.

   The **Service Control** message box is displayed. After a few seconds it disappears, and the **Status** field entry for Integration Server is changed to **Started**.

Integration Server Startup Switches

Table 1 lists the switches and configuration parameters that you use to change the default settings when starting Integration Server from the command line, when adding startup switch information to the `startup.bat` file, or when storing startup switch information in the `eis.cfg` file.

Table 1: Essbase Integration Server Switches

-?	-E	-N	-T
-A	-F	-P	-U
-B	-I	-Q	
-C	-L	-R	
-D	-M	-S	

For information on storing startup switches in `startup.bat` or `eis.cfg`, see "Storing Startup Switch Information in the Configuration File" on page 32 and "Storing Startup Switch Information in startup.bat (Windows Only)" on page 34.

The following sections are descriptions for the default switches.

-?

To view a list of available switches for starting Integration Server, type -? at a command prompt; for example, type:

```
olapisvr -?
```

-A

To ignore the parent of a given member when updating OLAP intersections during a drill-through operation.

If you want to ignore parents when updating OLAP intersections, do one of the following:

- In Windows, add -AY to the `startup.bat` file.
- In UNIX, type -AY when you start Integration Server.
- In Windows or UNIX, add `[A]=Y` to the `eis.cfg` file.

By default, Integration Server sets the value at N so that parents are not ignored when updating OLAP intersections. If the parent is ignored during an update of OLAP intersections, subsequent drill-through operations cannot be performed on any shared members related to that parent.

In Windows:

In the `startup.bat` file, to ignore the parent of a given member when updating OLAP intersections during a drill-through operation, add -IY after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -AY
```

In UNIX:

To ignore the parent of a given member when updating OLAP intersections during a drill-through operation, type:

```
olapisvr -AY
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[A]=Y
```

-B

To set the fetch size that Integration Server uses when transferring rows from the data source to the Analytic Services database, do one of the following:

- In Windows, add `-B`*`numberOfRows`* to the `startup.bat` file.
- In UNIX, type `-B`*`numberOfRows`* when you start Integration Server.
- In Windows or UNIX, add `-B`*`numberOfRows`* to the `eis.cfg` file

The default number is 100 rows; the range is from 1 to 32767 rows.

To maximize performance, estimate the maximum size in bytes of the rows being loaded from the data source. Divide 30,000 by that estimated size to determine the optimal number of rows to process each time. The goal is to generate blocks of data that do not exceed 30 KB. If the size of a block of rows exceeds the 30 KB threshold, the load could fail.

In Windows:

In the `startup.bat` file, to set the number of rows to 150, add -B150 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -B150
```

In UNIX:

For example, to set the number of rows to 150, type:

```
olapisvr -B150
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[B]=150
```

-C

To set the number of records that Integration Server commits to Analytic Services during a data load, do one of the following:

- In Windows, add `-C`*`number of records`* to the `startup.bat` file.
- In UNIX, type `-C`*`number of records`* when you start Integration Server.
- In Windows or UNIX, add `-C`*`number of records`* to the `eis.cfg` file.

The default setting is all records. Notice that the -C switch can be used concurrently with the -N switch.

When used in conjunction with the DATAERRORLIMIT setting in the `essbase.cfg` file, you can maximize the number of records written to `dataload.txt`, the data load error file. For example, if you set DATAERRORLIMIT to 65,000, the maximum number of error records allowed in the Analytic Services log, and then set -C to 65,000 or less, you will be able to view all error records allowed by Analytic Services.

The recommended setting for testing purposes is 5,000 to 10,000 records. After you have completed data load testing, shut down Integration Server to reset the number of records to the default setting of all records or reset to the limit allowed by DATAERRORLIMIT. If you have included a setting for testing purposes for this startup switch in `eis.cfg`, be sure to edit the file to clear the setting back to all records.

Setting a smaller number of records during testing, such as after every 5,000 records have been transferred to Analytic Services, enables you to see data load errors incrementally. Viewing the errors provides you an opportunity to correct data source problems before continuing, instead of waiting for an entire data load to finish before taking corrective actions. Resetting the number of records to the default of all records results in faster performance. Error records, however, are viewable only at the completion of the data load.

There is no limit to the number of records that Integration Server can transfer to Analytic Services during a data load. Analytic Services can use all available memory resources allocated to it. If, however, you experience memory problems during a data load, set **-**C to a smaller value.

Note: This setting does not apply to member loads.

In Windows:

In the `startup.bat` file, to set the number of records that Integration Server incrementally transfers to Analytic Services during a data load to 5,000, add **-**C5000 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -C5000
```

In UNIX:

For example, to set the number of records that Integration Server incrementally transfers to Analytic Services during a data load to 5,000, type:

```
olapisvr -C5000
```

In Windows or UNIX:

For example, to set the number of records that Integration Services incrementally transfers to Analytic Services during a data load to 5,000, in the `eis.cfg` file, type:

```
[C]=5000
```

-D

To specifiy whether to add the DISTINCT clause to the SELECT statement of drill-through SQL.

The default behavior is to add the DISTINCT clause. If you do not want to add the DISTINCT clause, do one of the following:

- In Windows, add -DN to the `startup.bat` file.
- In UNIX, type -DN when you start Integration Server.
- In Windows or UNIX, add -DN to the `eis.cfg` file.

In Windows:

In the `startup.bat` file, to specify that the DISTINCT clause should not be added to drill-through queries, add -DN after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -DN
```

Any other value or no value specifies that the DISTINCT clause should always be added to drill-through SQL.

In UNIX:

To specify that the DISTINCT clause should not be added to drill-through queries, type:

```
olapisvr -DN
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[D]=N
```

-E

To give the log file a different name, do one of the following:

- In Windows, add `-E`*`log_file_name`* to the `startup.bat` file.
- In UNIX, type `-E`*`log_file_name`* when you start Integration Server.
- In Windows or UNIX, add `-E`*`log_file_name`* to the `eis.cfg` file.

The `.log` extension is added automatically. The default name and location is *`ISHOME`*`\log\olapisvr.log`.

In Windows:

In the `startup.bat` file, to name the log file `myserver.log` and put it in the `temp` directory on Windows, add `-Ec:\temp\myserver` after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe -E"c:\temp\myserver
```

Do not type the `.log` file extension. The `.log` extension is automatically appended to the name you type for the log.

Always enclose path names that contain spaces with quotation marks (`""`); for example, type:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -I -E"c:\program files\eis\myserver"
```

In UNIX:

To name the log file `myserver.log` and put it in the `temp` directory on Windows, type:

```
olapisvr -Ec:\temp\myserver
```

Enclose path names that contain spaces with quotation marks (`""`); for example, type:

```
olapisvr -E"c:\program files\eis\myserver"
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[E]=c:\temp\myserver
```

or

```
[E]=c:\program files\eis\myserver
```

Note: In the `eis.cfg` file, you do not need to type quotation marks around path names that contain spaces.

-F

To specify whether to add parentheses around user-defined drill-through filters. See Essbase Spreadsheet Add-in help for information on drill-through filters.

The default behavior is not to add parentheses around user-defined drill-through filters. If you want to add parentheses, do one of the following:

- In Windows, add -FY to the `startup.bat` file.
- In UNIX, type -FY when you start Integration Server.
- In Windows or UNIX, add -FY to the `eis.cfg` file.

In Windows:

In the `startup.bat` file, to specify that parentheses should be added around user-defined drill-through filters, add -FY after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -FY
```

Any other value or no value specifies that parentheses should not be added around user-defined drill-through filters.

In UNIX:

To specify that parentheses should be added around user-defined drill-through filters, type:

```
olapisvr -FY
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[F]=Y
```

-I

To prevent Integration Server from shutting down as a service on Windows after you log off, add -I to the `startup.bat` file.

When you run Integration Server as a service on Windows, it is highly recommended that you use the -I parameter.

Note: This configuration parameter is applicable only in Windows environments.

Note: The -I parameter cannot be entered in the eis.cfg file. It can only be specified in the startup.bat file.

In Windows:

In the `startup.bat` file, to prevent Integration Server from shutting down as a service, add -I after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -I
```

-L

To set the level of detail of the messages that Integration Server logs, do one of the following:

- In Windows, add `-L`*`level`* to the `startup.bat` file.
- In UNIX, type `-L`*`level`* when you start Integration Server.
- In Windows or UNIX, add `-L`*`level`* to the `eis.cfg` file.

The default level is 2.

You can specify the following levels:

0	to log all debug messages and the ODBC SQL generated by Integration Server; generate the `dataloadrecord.txt` file, listing the first 1,000 records, which may include successfully loaded records and rejected records; and create the `hisdld.rul` file consisting of the names of the dimensions loaded followed by the leaf node number of each dimension or member
2	to log all informational messages and some of the ODBC SQL generated by Integration Server
3	to log all warnings
4	to log all non-critical errors
7	to log all critical errors where Integration Server terminates the command
8	to log all critical server errors

The recommended level settings are 2 or 3.

In Windows:

In the `startup.bat` file, to use the highest level of logging, add -L0 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -L0
```

In UNIX:

To use the highest level of logging, type:

```
olapisvr -L0
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[L]=0
```

-M

To specify the location of the Analytic Services message database file if the file is moved to a new location, do one of the following:

- In Windows, add `-M`*`path\essbase.mdb`* to the `startup.bat` file.
- In UNIX, type `-M`*`path\essbase.mdb`* when you start Integration Server.
- In Windows, add `-M`*`path\essbase.mdb`* to the `eis.cfg` file.

The default location is *ARBORPATH*`\bin\essbase.mdb`.

In Windows:

In the `startup.bat` file, to specify the new location of the Essbase message database file as `essbase71\bin`, add `-Mc:\essbase71\bin\essbase.mdb` after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -Mc:\essbase71\bin\essbase.mdb
```

Always enclose path names that contain spaces with quotation marks (`""`); for example, type:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -M"c:\ess 71\bin\essbase.mdb"
```

In UNIX:

To specify the new location of `essbase.mdb`, type:

```
olapisvr -Mc:\essbase71\bin\essbase.mdb
```

Enclose path names that contain spaces with quotation marks (" "); for example:

```
olapisvr -M"c:\ess 71\bin\essbase.mdb"
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[M]=c:\essbase71\bin\essbase.mdb
```

or

```
[M]=c:\ess 71\bin\essbase.mdb
```

Note: In the `eis.cfg` file, you do not need to add quotation marks around path names that contain spaces.

-N

To specify the number of threads Integration Server uses when sending data to Analytic Services during a data load, do one of the following:

- In Windows, add `-N`*`number_of_threads`* to the `startup.bat` file.
- In UNIX, type `-N`*`number_of_threads`* when you start Integration Server.
- In Windows or UNIX, add `-N`*`number_of_threads`* to the `eis.cfg` file.

The default setting is 1 thread.

The -N switch may be used concurrently with the -C switch.

The number of threads allocated to the Integration Server data load optimization process is controlled by the -N switch. Typically, users start with 2 threads and adjust this number based on testing in their environment.

In Windows:

In the `startup.bat` file, to set the number of threads to 2, add -N2 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -N2
```

In UNIX:

To set the number of threads to 2, type:

```
olapisvr -N2
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[N]=2
```

-P

To set the TCP port number with which Integration Server communicates with its clients to a port different from the default 3388, do one of the following:

- In Windows, add `-P`*`portnumber`* to the `startup.bat` file
- In UNIX, type `-P`*`portnumber`* when you start Integration Server
- In Windows or UNIX, add `-P`*`portnumber`* to the `eis.cfg` file

In Windows:

In the `startup.bat` file, add `-P`*`portnumber`* after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -P8850
```

If you set the port number to a value other than the default, in Integration Services Console you must log in to the Integration Server by typing the server name or IP address and the non-standard port number, separated by a colon; for example:

```
aspen:8850
```

When starting Integration Services Shell, to set the TCP port number to a port different from the default, at the command line, type `-P`*`portnumber`*. For example:

```
olapicmd -P8850
```

In UNIX:

When starting Integration Server or Integration Services Shell, to set the TCP port number to port different from the default, type:

```
olapisvr -P8850
```

or

```
olapicmd -P8850
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[P]=8850
```

-Q

To specify the maximum number of blocks in the data load queue when performing a data load, do one of the following:

- In Windows, add `-Q`*`maximum_number_of_blocks`* to the `startup.bat` file.
- In UNIX, type `-Q`*`maximum_number_of_blocks`* when you start Integration Server.
- In Windows or UNIX, add `[Q]=`*`maximum_number_of_blocks`* to the `eis.cfg` file.

The default setting is 64 blocks.

In Windows:

In the `startup.bat` file, to set the maximum number of blocks in the data load queue to 128, add -Q128 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -Q128
```

In UNIX:

To set the maximum number of blocks in the data load queue to 128, type:

```
olapisvr -Q128
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[Q]=128
```

-R

To specify that users are restricted from browsing table records when customizing a drill-through report.

- When the [R] parameter is not set, users have unrestricted access to all table records.
- When the [R] parameter is set to Y, users have no access to any table records.
- When the [R] parameter is set to F, users have no access to fact table records but have unrestricted access to all other table records.

To restrict users from browsing fact table records, do one of the following:

- In Windows, add `-R`*`access_parameter`* to the `startup.bat` file.
- In UNIX, type `-R`*`access_parameter`* when you start Integration Server.
- In Windows or UNIX, add `[R]=`*`access_parameter`* to the `eis.cfg` file.

By default, the [R] parameter is not set, giving users unrestricted access to all table records.

In Windows:

In the `startup.bat` file, to prevent users from having any access to any table records, add -RY after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -RY
```

In the `startup.bat` file, to prevent users from having access to fact table records, add -RF after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -RF
```

In UNIX:

To prevent users from having any access to any table records, type:

```
olapisvr -RY
```

To prevent users from having any access to fact table records, type:

```
olapisvr -RF
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[R]=Y
```

or

```
[R]=F
```

-S

To specify the stack size in megabytes of each thread that Integration Server creates, do the following:

- In Windows, add `-S`*`number_of_threads_created`* to the `startup.bat` file.
- In UNIX, type `-S`*`number_of_threads_created`* when you start Integration Server.
- In Windows or UNIX, add `[S]=`*`number_of_threads_created`* to the `eis.cfg` file.

For Windows and Solaris, no stack size setting is required. For AIX, the default stack size setting is 3 MB; for HP-UX, the default stack size setting is 8 MB.

In Windows:

In the `startup.bat` file, to set the number of threads that Integration Server creates to 1, add -S1 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -S1
```

In UNIX:

To set the number of threads that Integration Server creates to 1, type:

```
olapisvr -S1
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[S]=1
```

-T

To set the number of network listeners that Integration Server starts, do one of the following:

- In Windows, add `-T`*`number`* to the `startup.bat` file
- In UNIX, type `-T`*`number`* when you start Integration Server
- In Windows or UNIX, add `[T]=`*`number`* to the `eis.cfg` file

The default number is 10. The recommended setting is 10.

Network listeners receive requests from Integration Services Console. Integration 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 Integration Server.

In Windows:

In the `startup.bat` file, to set the number of network listeners that Integration Server starts to 17, add -T17 after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -T17
```

In UNIX:

To set the number of network listeners Integration Server starts to 17, type:

```
olapisvr -T17
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[T]=17
```

-U

To specify whether the Analytic Services database and application should be unloaded from memory after a load is completed.

Integration Server can successfully unload the Analytic Services database from memory only if there are no users accessing it.

To specify whether the Analytic Services database and application should be unloaded from memory after a load is completed, do one of the following:

- In Windows, add -UY to the `startup.bat` file.
- In UNIX, type -UY when you start Integration Server.
- In Windows or UNIX, add `[U]=Y` to the `eis.cfg` file.

By default, Integration Server does not unload the Analytic Services database from memory after a load is completed.

In Windows:

In the `startup.bat` file, to specify whether the Analytic Services database and application should be unloaded from memory after a load is completed, add -UY after `"C:\Hyperion\EIS\bin\olapisvr.exe"`. For example:

```
"C:\Hyperion\EIS\bin\olapisvr.exe" -UY
```

In UNIX:

To specify whether the Analytic Services database and application should be unloaded from memory after a load is completed, type:

```
olapisvr -UY
```

In Windows or UNIX:

In the `eis.cfg` file, this parameter is specified in the following format:

```
[U]=Y
```

Storing Startup Switch Information in the Configuration File

The `eis.cfg` file is used for two purposes:

- As an environment for Teradata and Oracle users to configure the required JDBC drivers, as discussed in the *Essbase Integration Services Installation Guide*.
- As a means for passing startup switches and configuration parameters, other than the defaults, when launching Integration Server

The `eis.cfg` file is a text file that resides in the `\bin` directory on Windows, UNIX, and Linux platforms. When you start Integration Server from the command line using the `startup.bat` command (in DOS or UNIX), the startup routine first checks the contents of `eis.cfg` for any startup switch or parameter information. After the check of `eis.cfg` contents, one of the following situations occurs:

- If you added startup switch or configuration parameter syntax to the `eis.cfg` file, those settings that you added override the default settings for those switches and parameters (see "Integration Server Startup Switches" on page 17 and "Storing Startup Switch Information in the Configuration File" on page 32).
- If you did not add startup switch or parameter information to `eis.cfg`, the default settings (see "Integration Server Startup Switches" on page 17 and "Storing Startup Switch Information in the Configuration File" on page 32) are used to start Integration Server.
- If `eis.cfg` contains startup switch syntax and you enter startup switch overrides in `startup.bat` any startup switches in `startup.bat` override both the startup switch settings contained in the `eis.cfg` file and the default settings.

➤ To store startup switch and parameter settings in the `eis.cfg` file:

1. Open the `eis.cfg` file in any text editor.
2. Add the startup switch that you want to store by typing the startup switch letter in brackets; for example:

   ```
   [B]
   ```

3. Add the new information for startup switches in the following format; for example:

   ```
   [L]=0
   [E]=c:\temp\myserver
   [N]=40
   [M]=c:\ess 65\bin\essbase.mdb
   ```

   To enter the above overrides in the `startup.bat` file, type:

   ```
   "C:\Hyperion\EIS\bin\olapisvr.exe" -I -L0
   -Ec:\temp\myserver -N40 -M"c:\ess 71\bin\essbase.mdb"
   ```

4. Save and close `eis.cfg`.

Storing Startup Switch Information in startup.bat (Windows Only)

The `startup.bat` file is used for two purposes:

- As an environment to configure the common Hyperion components, such as Java, ODBC, and JDBC drivers, as discussed in the *Essbase Integration Services Installation Guide*.
- As a means for passing startup switches, other than the defaults, when launching Integration Server

The `startup.bat` file is a text file which resides in the `\bin` directory on Windows, UNIX, and Linux platforms. When you start Integration Server from the command line (in DOS or UNIX) or as a Windows service, the startup routine first checks the contents of `startup.bat` for any startup switch information. After the check of `startup.bat` contents, the startup routine checks `eis.cfg` for any startup switch and configuration parameter information. Then one of the following situations occurs:

- If you added startup switch syntax to the `startup.bat` file, those settings that you added override the default settings for those switches and any startup switch information you entered in `eis.cfg` (see "Integration Server Startup Switches" on page 17).
- If you added startup switch or configuration parameter syntax only to the `eis.cfg` file, those settings that you added override the default settings for those switches and parameters (see "Integration Server Startup Switches" on page 17 and "Storing Startup Switch Information in the Configuration File" on page 32).
- If you did not add startup switch or parameter information to `startup.bat` or `eis.cfg`, the default settings (see "Integration Server Startup Switches" on page 17 and "Storing Startup Switch Information in the Configuration File" on page 32**)** are used to start Integration Server.

➤ To store startup switch settings in the `startup.bat` file:

1. Open the `startup.bat` file in any text editor.
2. If you want to change any default startup switches, type the command syntax in the line that calls the Integration Server.

   The values for startup switches that you enter into `startup.bat` override any startup switch values you have stored in the `eis.cfg` file.

   When entering parameters for startup switches in `startup.bat`, enclose path names that contain spaces in quotation marks (`""`). The following example illustrates the syntax for passing the -L, -E, -M, -N options:

   ```
   "C:\Hyperion\eisserv\bin\olapisvr.exe" -I -L0
   -Ec:\temp\myserver -N40 -M"c:\ess 65\bin\essbase.mdb"
   ```

   Note: In the `startup.bat` file, the above syntax should be entered all on one line.

   You can only enter startup switch information into `startup.bat`. Configuration parameters must be entered in `eis.cfg`.
3. Save and close `startup.bat`.

Reconnecting When Connections Are Lost

If you lose the connection to Integration Server and OLAP Metadata Catalog, you can reconnect directly from Integration Services Console without losing any unsaved work.

➤ To reconnect to Integration Server:

1. Start Integration Server using one of the methods described in “Starting Integration Server” on page 14.
2. In the Integration Services Console, select **Connections** > **OLAP Metadata Catalog** > **Reconnect**.

   Note: The connection is made implicitly; no messages are displayed.

Shutting Down Integration Server

When you finish using Integration Server, shut it down by using any of the following methods.

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

➤ To shut down Integration Server from the command line, either close the Integration Server window, or use Integration Services Shell.

- On Windows platforms, close the Integration Server window.
- Issue the SHUTDOWN command in the Integration Services Shell.

  For more information, see "Integration Services Shell Commands" on page 109.

➤ To shut down Integration Server from the Windows desktop:

1. Right-click the button representing Integration Server in the Windows task bar.
2. Select **Close**.

➤ To shut down Integration Server as a Windows service:

1. Select **Start** > **Settings** > **Control Panel**.
2. Open the **Services** dialog box.
   - On Windows 2000, double-click **Administrative Tools**, and then double-click **Services**.
   - On Windows XP, select **Performance & Maintenance**, and then **Administrative Tools**. In the **Administrative Tools** window, double-click **Services**.
3. Locate **Essbase Integration Server** in the list of services.
4. Select, but do not double-click, **Essbase Integration Server**.

5. Stop the service.

   - On Windows 2000, select **Action** > **Stop**.
   - On Windows XP, right-click **Essbase Integration Server** and select **Stop**.

   The **Service Control** message box is displayed. After a brief delay, it disappears and the **Status** field for **Essbase Integration Server** is changed, and Integration Server is stopped.

Integration Server Log File

By default, the Integration Server log file is named `olapisvr.log` and is located in the *`ISHOME`*`\log` directory. You can view it from Integration Services Console by selecting the Tools > View Log File command to launch the Server Log file dialog box. You can also view the log by opening `olapisvr.log` in any text editor. To create the log file with a different location or name, use the -E switch. For more information, see "Integration Server Startup Switches" on page 17.

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

- Processing messages
- Member load error messages
- The settings for switches used to start Integration Server from the command line or as a Windows service
- Two copies of each SQL statement generated to access the data source as follows:
  - The first SQL statement is in the SQL dialect understood by the ODBC driver and is generated by Integration Server.
  - The second SQL statement is generated by the ODBC driver and is translated by the driver into the dialect of SQL understood by the data source being accessed.
- The 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 that is running
- Other informational messages

Additionally, when a user accesses a drill-through report using Essbase Spreadsheet Add-in, the following items related to the drill-through report are logged in the Integration Server log file:

- The name of the drill-through report
- The name of the metaoutline with which this drill-through report is associated
- The name of the Analytic Server computer that is being accessed
- The Analytic Services application name and database name
- The user ID of the user accessing this drill-through report

For more information on setting the logging level, see "Integration Server Startup Switches" on page 17.

Integration Server error message text is located in `ISHOME\bin\error.txt` on Windows systems; `$ISHOME/bin/error.txt` on UNIX.

Analytic Services error message text is located in `ISHOME\bin\message.txt` on Windows systems; `$ISHOME/bin/message.txt` on UNIX.

The Data Load Error File

In the event of errors during a data load, a file named `dataload.txt` is placed in a folder that Integration Server creates for the load under the `ISHOME/loadinfo` directory. The `dataload.txt` file lists rejected dimensions and members along with error message codes that help to identify the problems that occurred during the data load.

The folder that Integration Server creates in the `loadinfo` directory is named in the following format:

```
<application_database_timestamp_sessionnumber>
```

If you performed a data load for the MyTBC application and MyTBC_DB database at 10 P.M. on May 1, 2003, the folder is named:

```
MyTBC_MyTBC_DB_2003_May_1_10_00_pm_<sessionnumber>
```

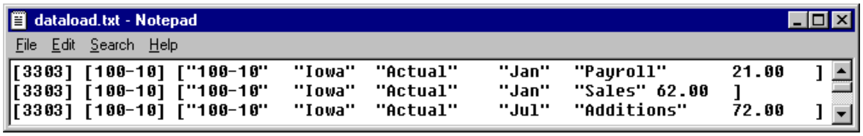
In the case of a data load failure, open the `dataload.txt` file located in the folder described previously and review the error codes listed in the first column of each record.

These are the most common error codes:

Message Number	Message
3303	Member not found in database.
3304	Insufficient access to store data.
3333	Bad data value supplied.
3335	Record rejected because of duplicate member names.
3336	Member/Data Unknown.
3337	Record rejected because of dimension conflicts with Header Name.

Figure 1 shows three records from the `dataload.txt` file with the error code 3303.

Figure 1: Example from dataload.txt File



```
[3303] [100-10] ["100-10"   "Iowa"   "Actual"    "Jan"   "Payroll"      21.00    ]
[3303] [100-10] ["100-10"   "Iowa"   "Actual"    "Jan"   "Sales" 62.00  ]
[3303] [100-10] ["100-10"   "Iowa"   "Actual"    "Jul"   "Additions"    72.00    ]
```

Unicode

Sharing data across national and language boundaries is a challenge for multi-national businesses. Traditionally, each computer stores and renders text based on its locale specification. A *locale* identifies the local language and cultural conventions such as the formatting of currency and dates, sort order of the data, and the character set encoding to be used on the computer. The *encoding* of a character set refers to the specific set of bit combinations used to store the character text as data, as defined by a code page or an encoding format. In Analytic Services, *code pages* map characters to bit combinations for non-Unicode encodings.

Because different encodings can map the same bit combination to different characters, a file created on one computer can be misinterpreted by another computer that has a different locale.

The Unicode Standard was developed to enable computers with different locales to share character data. Unicode provides encoding forms with thousands of bit combinations, enough to support the character sets of multiple languages simultaneously. By combining all character mappings into a single encoding form, Unicode enables users to correctly view character data created on computers with different locale settings.

Users whose computers are set up in different languages can work with the same database. For example, using alias tables in their respective languages, users in Taiwan can view database reports displaying Chinese characters while users in France can view the same reports in French characters.

User-defined character sets (UDC) are not supported and the Chinese National Standard GB 18030-2000 is not supported. Unicode-mode applications do not support the hybrid analysis, query logging, triggers, and data mining features. Unicode-mode applications also do not support the MaxL Data Manipulation Language (MaxL DML). SQL Interface does not work with Unicode-mode applications.

Note: For additional information on using Unicode in Essbase Services, see the *Essbase Analytic Services Database Administrator's Guide*.

For additional information about the Unicode Standard, see `www.unicode.org`.

Unicode and Non-Unicode Application Modes

Applications are designated as Unicode-mode applications or non-Unicode-mode applications.

Unicode-mode applications support multiple character sets. When it works with Unicode-mode applications, Integration Services uses the UTF-8 encoding form to interpret and store character text. Character-based objects in Unicode-mode applications, such as member and alias names, can include characters from different languages.

Clients working with Unicode-mode applications can have different locales than Analytic Server. For example, client computers with Japanese locales and client computers with German locales can work with the same Unicode-mode application on an Analytic Server that has a Spanish locale.

For Unicode-mode applications, the limits of most object names are longer than those names in non-Unicode-mode applications and the limits are calculated based on characters rather than bytes. See "Increased Name Lengths in Unicode" on page 42.

Non-Unicode-mode applications support one character set that is defined by the a locale value which must be the same for Analytic Server and all non-Unicode clients that work with the non-Unicode-mode applications. Both Unicode-mode and non-Unicode-mode applications can reside on the same Analytic Server.

Note: You cannot convert a Unicode-mode application to non-Unicode mode.

When to Use Unicode-Mode Applications

Consider working with Unicode-mode applications only if you have any of the following situations:

- You need to enable users with different languages to view, in their own languages and character sets, information from a common database. For example, using alias tables in Japanese and German, users in Japan and Germany could view, in their own languages, information about a common product set.
- You need to handle object names longer than non-Unicode-mode applications support. For example, application and database names need to be larger than eight characters or, if you are working with a multi-byte character set, you need to handle more characters in object names.
- You have experienced what is called the "round-trip" problem. The round-trip problem can occur in communications between multi-byte operating systems and application programs where two different bit values can map to the same character. As Java applications, Administration Services and Deployment Services always work in Unicode. No encoding conversions occur when these clients work with Unicode-mode applications and UTF-8-encoded text files; hence no round-trip conversion errors.

When deciding on using Unicode-mode applications, you should also consider the following points:

- Using non-Unicode text files with Unicode-mode applications requires an understanding of locales and care in managing to them. To prevent errors that could cause database corruption, using UTF-8-encoded files is recommended. For details, see the *Essbase Analytic Services Database Administrator's Guide*.
- To work with Unicode-mode applications, custom client applications that were written to support non-Unicode-mode applications must be built to use the longer string lengths used by Unicode-mode applications. This may be a simple re-build or may involve re-programming, depending on the design of the applications. Also, depending on how they are coded, the new client applications may require more memory.

Increased Name Lengths in Unicode

For Unicode-mode applications, the maximum number of characters allowed in strings such as application, database, and member names is greater than the maximum allowed for non-Unicode-mode applications.

For non-Unicode-mode applications, the maximum length of most object names is calculated in bytes. For Unicode-mode applications, the maximum length of most object names is calculated based on the number of characters, regardless of how many bytes each character requires. Not limiting by bytes is advantageous for applications using multi-byte character sets, such as Chinese and Japanese. For example, member names in Unicode-mode applications can contain 80 characters, even if they are multi-byte characters.

Note: The increase in size limits does not affect the size of the outline, but may affect the size of user-written client programs.

Unicode-Enabled Administration Tools

Hyperion provides Administration Services and MaxL to administer Unicode-mode applications. The main administration activities include, in addition to the normal Analytic Services administration activities, changing the Unicode-related mode of the Analytic Server to enable or disable creation of Unicode-mode applications, creation of Unicode-mode applications, migration of non-Unicode-mode applications to Unicode mode, and viewing the Unicode-related status of servers and applications.

Unicode-Enabled API

Without recompilation, existing custom-written client programs are not Unicode-enabled. These programs use short strings and short buffers. You can continue to use these programs with non-Unicode-mode applications.

Depending on how they are written, in order to provide restricted access to Unicode-mode applications, existing custom-written client programs can be recompiled in a Unicode-enabled release of Analytic Services. Simply recompiled, these programs work with long buffers but short strings.

For complete access to Unicode-mode and non-Unicode-mode applications, existing custom-written applications need to be modified using the new Analytic Services API functions for Unicode. Rewritten and compiled clients work with long buffers and long strings for full Unicode support. For information about updating custom-written client programs, see the *API Reference*.

Spreadsheet Retrieval in Unicode

The Analytic Services Spreadsheet Add-in for Excel supports viewing data in both Unicode and non-Unicode-mode applications. Older versions of the spreadsheet add-ins for Excel and Lotus 1-2-3 can view data only in non-Unicode-mode applications. The older versions of these add-ins are available only through older releases of Analytic Services that are not Unicode-enabled.

Setting Up Computers for Unicode Support

Consider setting up the following non-Hyperion tools for working with UTF-8 encoded text on computers that will manage Unicode-mode applications.

- To enable viewing UTF-8 encoded text that contains non-ASCII characters, install a font that supports UTF-8 encoding.
- To support manual editing of data sources or other text files, install a Unicode editor that enables viewing, creating, and modifying UTF-8 encoded files and includes the UTF-8 signature. You can also use this type of editor to read UTF-8-encoded files and save them in non-Unicode encoding. A Unicode editor is not necessary. Analytic Services provides the Analytic Services Unicode File Utility which converts text files to UTF-8 encoding.

Defining Password Security

Analytic Services security is defined at the Analytic Server level. To create passwords, use characters encoded according to the code page specified in the ESSLANG value on Analytic Server. In a multiple character set environment, consider using the characters in the standard ASCII character set, in the range from 0 through 127; these characters are common to all code pages as well as UTF-8.

Using Control Characters in Report Scripts

When working with reports for non-Unicode-mode applications, Report Writer uses language-specific codes that enable columnar data to line up precisely. Unicode does not contain these language-specific formatting codes within its encoding. As a result, report columns may not line up precisely.

Non-Unicode-mode applications using single-byte code pages continue to support the following Hexadecimal control characters in report scripts: x20, x09, x0A, x0D, xB3, xC3, xC4, xC2, xC0, x0C, xB1.

Unicode-mode applications and non-Unicode-mode applications using single-byte code pages support the following Hexadecimal control characters in report scripts: x20, x09, x0A, x0C, x0D.

Working with Logs

By default, the agent log is encoded to the locale specified by the ESSLANG variable defined for Analytic Server. For object names such as application and database names, be sure to use characters that are included in the code page specified by the ESSLANG locale. Otherwise, Analytic Server may display unrecognizable characters in the agent log.

You can use the configuration setting, UNICODEAGENTLOG, to tell Analytic Server to write the agent log in UTF-8 encoding. In UTF-8 encoding, with a font that supports it, all characters should be readable. For details about the UNICODEAGENTLOG configuration setting, see the *Technical Reference*.

Application and outline change log files for databases in Unicode-mode applications are UTF-8 encoded. You can use Log Viewer in Administration Server or a UTF-8 editor to view application logs. You need a UTF-8 editor or viewer to view outline change logs. You cannot change to non-Unicode encoding the encoding of logs related to Unicode-mode applications.

Essbase Integration Services for SAP BW

Integration Services supports SAP Business Information Warehouse (BW) data collection. This separately purchased and licensed option is known as “Essbase Integration Services for SAP BW.”

Integration Services functions with SAP BW through Microsoft OLE Database (OLE DB) for online analytic processing (OLAP). The open database connectivity (ODBC) driver used for this process is manufactured by Simba Technologies. (See the *Hyperion Connect Installation Guide*.)

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:

- Integration Services users: these users can access OLAP models and metaoutlines stored in OLAP Metadata Catalogs. Integration Services users are the database users defined in the data source that contains the OLAP Metadata Catalog you are using.
- Analytic Services users: these users can access the Analytic Services database that you create from a metaoutline.
- Source database and data warehouse users: these users can access the tables and views in the specified source database or data warehouse.

Use the tools provided with each data source to manage each set of users. Consult the data source 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 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 Integration Services and are completely separate from the locking performed by the data source database.

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, Integration Services gives you a read lock on that OLAP model. When you open a metaoutline in standard access mode, 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, Integration Services upgrades the read lock to a write lock for the duration of the save. After 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 49.

➤ 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 Console Help.
- 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 Integration Services Users with Locks" on page 50.

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, 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, 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 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 Integration Services Users with Locks" on page 50. 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 Integration Services Users" on page 52.

Viewing 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 data source 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 2004 4:43PM
sa           889845263          2               1         Apr 9 2004 4:43PM
sa           889844639          1               1         Apr 9 2004 5:20PM
sa           892167813          2               1         Apr 9 2004 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 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 Integration Services Users with Locks" on page 50.

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 data source:

   ```
   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. 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 Integration Services Console. For more information, see the Integration Services Console Help.
- Save the metaoutline with a different name by selecting File > Save As. For more information, see the Integration Services Console Help.
- To edit tables containing the permissions for the OLAP model or metaoutline, use the tools provided with the data source that contains OLAP Metadata Catalog. For more information on the tables to edit, see "Viewing Permissions for OLAP Models" on page 53 or "Viewing Permissions for Metaoutlines" on page 54.

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 data source 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 data source.

   ```
   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 data source.

```
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 data sources and load data into Essbase Analytic Services 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 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.

Note: If you are accessing SAP BW data sources, the latest troubleshooting information may be found in the Essbase Integration Services for SAP BW readme file.

This chapter contains the following topics:

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 Microsoft SQL Server) or synonyms for the tables (for DB2 and Oracle). For more information, see the *Essbase Integration Services Installation Guide*.
- Are all the following required components up and running?
    - Essbase Integration Server
    - The database servers that manage OLAP Metadata Catalog and the data source databases
    - The data source database listener for OLAP Metadata Catalog and the data source
- Is the database client software installed and configured on the computer where 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 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 Integration Server is installed?

    On UNIX, use `ivtestlib` 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 Integration Server?

  - Does the ODBC data source name (DSN) configuration identify the name of the data source 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 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 Integration Server computer? Do not assume that the datasource 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 data source 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 Analytic Services 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 88.

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

If none of the above steps help, review the list of database-specific problems in “Database-Specific Problems” on page 60.

Database-Specific Problems

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

- **DB2:** You may be required to bind the database driver to a DB2 server that is managing the databases you want. For more information, see the DB2 installation documentation.
- **Oracle OCI:** If connection attempts using Oracle OCI fail, be sure that when logged in as the Integration Services user (the user who installed Integration Services) on the Integration Server, you can successfully connect to the Oracle database using SQL*Plus. The name for the database that works in SQL*Plus should work in Essbase Integration Services when referred to as Oracle:Net Service Name (if you are using tnsnames) or Oracle:oname (if you are using Oracle onames).
- **SQL Server:** If you try to access an SQL Server database with the Microsoft-supplied 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.

If none of the previous suggestions help, proceed to “About ODBC and Connection Problems” on page 61 for an overview of connection problems, then see “Isolating Connection Problems” on page 62.

About ODBC and Connection Problems

There are three layers of software components that Integration Services Console must pass through to contact a data source: 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 data source.

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

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

- Software component problems:
  - The components of Integration Services or the data source 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.

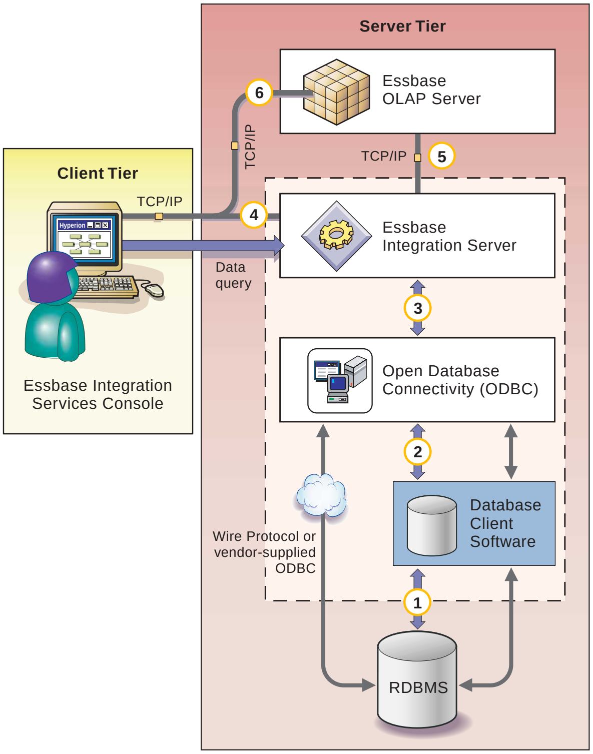
- Network and security problems:
  - A physical network path does not exist between the computers that must communicate with each other.
  - One or both computers attempting to communicate do not have the TCP/IP network protocol installed or configured.
  - The computer attempting to contact another 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. For example, this is the case when client and server computers are separated by a firewall.

Isolating Connection Problems

If the list under "Common Connection Problems" on page 58 and your own knowledge fail to diagnose the connection problem, the recommended strategy is to test the components and connections starting at the data source and backtracking through the connections to Integration Services Console.

The client software must pass through three layers of components to connect to the data source containing OLAP Metadata Catalog and the data source, as illustrated in Figure 2.

Figure 2: Essbase Integration Services Components and Connections


Server Tier
Essbase OLAP Server
6
TCP/IP
TCP/IP
5
Client Tier
TCP/IP
4
Essbase Integration Server
Data query
3
Open Database Connectivity (ODBC)
2
Database Client Software
Wire Protocol or vendor-supplied ODBC
1
RDBMS
Essbase Integration Services Console

The numbers in Figure 2 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 data source (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 78.

➤ To isolate connection problems:

1. Attempt to connect to the data source databases using a database client utility program. For more information, see “Testing the Database Client Connection to the Data Source” on page 64.
2. Attempt to connect to the data source databases using ODBC. For more information, see “Testing the ODBC Connection to the Data Source” on page 66.

3. Attempt to connect to the data source databases using Integration Server. For more information, see "Testing the Integration Server Connection to the Data Source" on page 71.
4. Attempt to connect to the data source databases using Integration Services Console. For more information, see "Testing the Integration Services Console Connection to the Data Source" on page 74.
5. Attempt to connect to Essbase Analytic Server using Integration Server. For more information, see "Testing the Integration Server Connection to Analytic Server" on page 75.
6. Attempt to connect to Analytic Server using Integration Services Console. For more information, see "Testing the Integration Server Connection to Analytic Server" on page 75.

Testing the Database Client Connection to the Data Source

If your ODBC connections require database client software, the first step toward isolating connection problems in Integration Services is to attempt to connect the database client software to the data source that you are using. After 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 Integration Server program.

➤ To test the database client connection to the data source:

1. Log on to the computer running Integration Server as the user who starts the 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 that the client software is installed and the required environment variables are set. For more information, see "Correcting Database Client Problems" on page 81.

3. Connect to the OLAP Metadata Catalog database in the data source 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 data source for the OLAP Metadata Catalog database, and carefully check user accounts with connection problems. For more information see, “Correcting Data Source Problems” on page 80.

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 data source. For more information, see “Correcting Data Source Problems” on page 80. Also, check that the version of the database client software you are using is compatible with the version of the data source. For more information, see “Correcting Database Client Problems” on page 81.

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

   For more information, see “Testing the Integration Services Console Connection to the Data Source” on page 74. If the connection fails, proceed to “Testing the ODBC Connection to the Data Source” on page 66.

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

6. Repeat step 3 through step 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 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 Data Source

After you have established that the database client software and data source are communicating properly (if applicable), the next step in isolating connection problems is to test the ODBC connection to the data source databases.

On Windows systems, use ODBC Administrator to test connections to the data source databases. In UNIX, use the `odbcconfig` utility to verify ODBC configuration. Manually inspect the ODBC configuration files using `ivtestlib` provided with Integration Services or by using DemoODBC (available from Hyperion Solutions Technical Support). You can also use third-party utilities included with your data source.

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

ODBC tracing can also be used to track down connection problems. For more information, see "Using ODBC Tracing" on page 88.

Testing ODBC on Windows Systems

➤ To test ODBC connections to the data source databases:

1. On the Windows desktop, select **Start** > **Settings** > **Control Panel** to open the **Control Panel** window.
2. In the **Control Panel** window, 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 data source created for the OLAP Metadata Catalog or the data source database, and click **Configure**.

   A data source configuration dialog box is displayed. Depending on the data source and the driver that 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 data source successfully.

   Note: To be accessible from other computers, ODBC data sources must be configured as System DSNs, not User DSNs.

5. If you cannot connect to the database, check the Integration Server name, database name, user name, and password information for accuracy, and make sure that you are using the correct ODBC driver for the data source.

   For more information, see "Correcting ODBC Problems" on page 83.

6. After you have successfully completed the preceding steps, try connecting to the data source from Integration Services Console.

   For more information, see "Testing the Integration Services Console Connection to the Data Source" on page 74.

   If the connection from Integration Services Console fails, proceed to "Testing the Integration Server Connection to the Data Source" on page 71.

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 data source utility 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 data source 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 Integration Server as the user who starts the 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 ODBC data sources, 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.

   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 data source 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.

Note: 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 Integration Server as the user who starts the 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**.

4. At the prompts, type a data source name, user name, and password.

   If no errors are returned, the connection is valid.

   In the event of errors, proceed to step 5 to add corrected DSN information.

5. 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.

6. Type **1** and press **Enter** to list DSNs and verify that the DSN you just added is included in the list.
7. Type **4** and press **Enter** to test the new ODBC connection.

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

➤ To inspect the login script files:

1. Log on to the computer that runs Integration Server as the user who starts the 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 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 83.

   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 Integration Services Console Connection to the Data Source” on page 74.

   If you cannot make a connection, proceed to “Testing the Integration Server Connection to the Data Source” on page 71.

➤ To inspect the `odbc.ini` file:

1. Log on to the computer that runs Integration Server as the user who starts the 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 Integration Services script in the login script for the user. For more information, see "Correcting Integration Server Problems" on page 85.

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 Integration Services are listed in the ODBC Data Sources section; for example

   ```
   [ODBC Data Sources]
   TBC_MD=Essbase Integration Serveices sample catalog
   TBC=Essbase Integration Services 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=
    $HYPERION_Home/common/ODBC/Merant/4.2/ARor819.so
   Description=my oracle source
   ServerName=mytnsServerName
   ```

For DB2:

```
[TBC_MD]
Driver=
 $HYPERION_Home/common/ODBC/Merant/4.2/ARdb219.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 are specified.

   For more information, see “Correcting ODBC Problems” on page 83.

Testing the Integration Server Connection to the Data Source

After verifying the connections between the ODBC, the database client, and the data source, the next step in isolating a connection problem is to connect to the data source databases from Integration Server. Test this connection by running Integration Services Shell on the same computer that runs Integration Server. For more information on the use and syntax of 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. Analytic 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 Integration Server connection to the data source:

1. Verify that Integration Server is running.

   If Integration Server is not running, start it. If Integration Server will not start, check that the Integration Services environment variables are set. For more information, see “Correcting Integration Server Problems” on page 85.

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

   ```
   olapicmd
   ```

   If Integration Services Shell does not start, make sure that the software is installed and that the Integration Services environment variables are set. For more information, see “Correcting Integration Server Problems” on page 85.

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

   ```
   login cypress
   ```

4. Check the release of 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 Analytic Services 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;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password"
   ```

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

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, 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 Integration Server Problems” on page 85 and “Correcting ODBC Problems” on page 83. Repeat step 4 to step 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;CODEPAGE=
   English_UnitedStates.Latin1@Binary;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 data source.

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, Integration Services Shell connects to OLAP Metadata Catalog and then to the Analytic 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 Integration Server Problems” on page 85 and “Correcting ODBC Problems” on page 83. Repeat step 7 to step 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 Integration Services Console to the data source by proceeding to “Testing the Integration Services Console Connection to the Data Source” on page 74.

Testing the Integration Services Console Connection to the Data Source

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

➤ To test the Integration Services Console connection to the data source:

1. Verify that Integration Server is running.

   If Integration Server is not running, start it. If Integration Server will not start, check that the Integration Services environment variables are set. For more information, see “Correcting Integration Server Problems” on page 85.

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

3. Connect to the 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 Integration Server computer name. Also verify that you specified the correct user name and password for the OLAP Metadata Catalog database on the data source. For more information, see "Correcting Integration Services Console Problems" on page 86.

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 Integration Server computer name. Also verify that you specified the correct user name and password for the data source database on the data source. For more information, see "Correcting Integration Services Console Problems" on page 86.

Testing the Integration Server Connection to Analytic Server

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

➤ To test the Integration Server connection to an Analytic Server:

1. Verify that Integration Server is running.

   If Integration Server is not running, start it. If Integration Server will not start, check to ensure that the Integration Services environment variables are set. For more information, see "Correcting Integration Server Problems" on page 85.

2. Verify that the Analytic Server is running.

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

   ```
   olapicmd
   ```

   If Integration Services Shell does not start, make sure that the software is installed and that the Integration Services environment variables are set. For more information, see "Correcting Integration Server Problems" on page 85.

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

   ```
   login cypress
   ```

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

   ```
   setsource "DSN=TBC;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password"
   ```

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

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

   ```
   setcatalog "DSN=TBC_MD;CODEPAGE=
   English_UnitedStates.Latin1@Binary;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 data source.

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

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

   Note: If you are using Integration Services Console on a client computer that is outside the firewall for your network, and you require access to an Integration Server and Analytic Server located inside the firewall for your network, be sure to use a name or an IP address for the Analytic Server that is accessible from both sides of the firewall.

8. Test the connection to the Analytic 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 Analytic Server is running. Verify the name of the metaoutline in Analytic Services. Also verify the name of the test application and database in Analytic Services. Repeat step 7 to step 8 of this procedure to retest the connection.

   After you can connect successfully to the Analytic Server, proceed to "Testing the Integration Services Console Connection to Analytic Server" on page 77.

Testing the Integration Services Console Connection to Analytic Server

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

1. Verify that Integration Server is running.
2. Verify that Analytic Server is running.
3. Start Integration Services Console.
4. From Integration Services Console, connect to the 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 Analytic 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 Analytic Server and the correct user name and password. For more information, see the Analytic Services documentation.

   If the Analytic 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.”

   Note: If you are using Integration Services Console on a client computer that is outside the firewall for your network, and you require access to an Integration Server and Analytic Server located inside the firewall for your network, be sure to use a name or an IP address for the Analytic Server that is accessible from both sides of the firewall.

Correcting Connection Problems

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

To correct system-level problems that 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 79.
- Do all 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 Integration Server computer
  - OLAP Metadata Catalog database
  - Data source database
  - User account for the Analytic 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 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 the following software components:

  - Integration Server
  - ODBC software
  - Database client software
  - Data source 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 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 that 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 Integration Services, where Integration Server, Analytic Server, and the data source are all run on a single computer, you must manage five user name and password sets and use them consistently:

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

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

- User account on the Integration Services client computer
- User account on the computer that runs Integration Server
- User account on the computer that runs the data source

- OLAP Metadata Catalog database in the data source
- Data source database
- User account on the computer that runs Analytic Server
- Analytic Server software

Adding aliases or synonyms to the data source 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 Data Source Problems

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

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

  Some data sources 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 data source server using ODBC and also may be required on the data source 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 can create the OLAP Metadata Catalog tables either automatically or by using the creation scripts provided with 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 data source 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.

- If the data source 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 data source server computer, separate from the database users you create.

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

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

Correcting Database Client Problems

If you have problems connecting to the data source 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 data source server and may also be required on the data source server computer.

- Is the database client software compatible with the data source?

  Earlier or later versions of the database client software may not work with the data source version that you are running. For more information on version compatibility, see the data source 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 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 Integration Server (typically, hyperion) is configured to run the database client software.

  On UNIX systems, a script file is usually provided by the data source 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 data source 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 data source-specific identification syntax. For more information about correcting user name and password problems, see "Correcting Data Source Problems" on page 80.

- 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:

– AIX and HP-UX: onsoctcp

– Solaris: ontlitcp

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

Correcting ODBC Problems

To correct problems using ODBC to connect to the data source, investigate the following possible causes:

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

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

- Is the ODBC driver compatible with the data source? Is the driver compatible with the operating system?

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

- Are the data source 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 data source-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 Data Source Problems” on page 80.

- 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 Integration Services installation provides a sample odbc.ini file, which may not include data source parameters required for the data source that you are using. For more information on required data source-specific ODBC driver parameters, see the ODBC documentation provided with Integration Services in Adobe Portable Document Format (PDF) in the ISHOME/odbc/merant42/docs directory. For examples of odbc.ini file configurations for the supported data source platforms, see the *Essbase Integration Services Installation Guide*.

- Can other applications connect to the data source by using ODBC?

  If you can identify another application that is successfully using ODBC to connect to databases on the data source, 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 data source vendors provide utilities to test ODBC connections. For information, see the documentation for the data source.

If the preceding checks do not enable you to correct the problems connecting from ODBC to the data source, try using ODBC tracing to isolate the problem. For more information, see “Using ODBC Tracing” on page 88.

Correcting Integration Server Problems

To correct problems connecting to the data source from Integration Server, investigate the following possible causes:

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

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

  - Integration Server software
  - Related base components (on UNIX, these components are included as part of Integration Server software)
  - DataDirect ODBC drivers (unless ODBC and drivers are already installed)

- Is Integration Server running? Is it configured properly?

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

  If 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 environment variables must be included in the login script of the user who starts Integration Server. For more information, see the *Essbase Integration Services Installation Guide*.

  - What release of Analytic Services is installed?

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

  - Is port 3388 in use?

    Integration Server uses port 3388 to communicate with the Integration Services client software. If this port is being used by another program, Integration Server 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 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 Integration Services Console Problems

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

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

- Is Integration Services Console installed? Are all required components installed?

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

  - Integration Services Console software
  - Related base components

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

  For the 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 data source 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 Data Source Problems" on page 80.

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

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

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

  A particular 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 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 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 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 Integration Server and Integration Services Console.
4. From 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 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 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 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 Integration Server and Integration Services Console.
5. From 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 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 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 Analytic Services database, Essbase Integration Services retrieves the information stored in OLAP Metadata Catalog to determine:

- What data to retrieve from the data source
- What operations to perform on the data
- How to load the data into the Analytic Services database

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 2, depending on what OLAP Metadata Catalog component you want to restore.

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

Table 2: 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 data source.

Tables Relating to the OLAP Model

Table 3 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 95
- “Tables Relating to Drill-Through” on page 97
- “Tables Relating to Hybrid Analysis” on page 98
- “Miscellaneous Tables” on page 98

Table 3: 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.
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.

Table 3: OLAP Model Tables (Continued)

Table Name	Description
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.
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_DESCRIPTIONS	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 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 92
- "Tables Relating to Drill-Through" on page 97
- "Tables Relating to Hybrid Analysis" on page 98
- "Miscellaneous Tables" on page 98

Table 4: 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_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 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 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 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: 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 in the OLAP Metaoutline main window of the Integration Services Console (refer to online help for information on creating drill-through reports). These reports enable Essbase Spreadsheet Add-in users to view data in the data source database that is not stored in the Analytic Services database.

Table 5 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 92
- "Tables Relating to the Metaoutline" on page 95
- "Tables Relating to Hybrid Analysis" on page 98
- "Miscellaneous Tables" on page 98

Table 5: Drill-Through Tables

Table Name	Description
DTR_ATTRIBUTES	Contains information about drill-through columns. This information includes the order in which 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.
MO_PROPERTIES	Contains information about drill-through filters. This information includes the Essbase user name and password, drill-through report name, and metaoutline name.

Tables Relating to Hybrid Analysis

You enable Hybrid Analysis in the OLAP Metaoutline main window of the Integration Services Console (refer to online help for information on enabling Hybrid Analysis). Hybrid Analysis enables you to access lower-level data stored in a data source database. This stored data, although not physically loaded into Analytic Services, is mapped to the appropriate Analytic Services hierarchies and is available to Spreadsheet Add-in users. Unlike drill-through data, which is displayed on a separate worksheet, Hybrid Analysis data is displayed seamlessly on the current worksheet in Spreadsheet Add-in reports.

Table 6 describes the table in OLAP Metadata Catalog that stores information about Hybrid Analysis. For information about the other tables in OLAP Metadata Catalog, see:

- "Tables Relating to the OLAP Model" on page 92
- "Tables Relating to the Metaoutline" on page 95
- "Tables Relating to Drill-Through" on page 97
- "Miscellaneous Tables" on page 98

Table 6: Hybrid Analysis Tables

Table Name	Description
MO_PROPERTIES	Contains information about Hybrid Analysis columns. This information includes system-generated ID numbers for the OLAP model; the Hybrid Analysis column; and column precision, scale, and nullability.

Miscellaneous Tables

Table 7 describes tables in OLAP Metadata Catalog that store miscellaneous information about OLAP models and metaoutlines and about Integration Services. For information about the other tables in OLAP Metadata Catalog, see:

- "Tables Relating to the OLAP Model" on page 92
- "Tables Relating to the Metaoutline" on page 95

- “Tables Relating to Drill-Through” on page 97
- “Miscellaneous Tables” on page 98

Table 7: 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 Integration Services software you have installed. This information includes the release number of 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 Integration Services Shell commands and scripts to create Essbase Analytic Services outlines and load Analytic Services databases.

If you schedule member or data loads using Integration Services Console, Integration Services automatically creates an 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 Integration Services Shell commands, see "Integration Services Shell Commands" on page 109.

Starting Integration Services Shell

Before you start 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 Integration Services Shell:

- A data source database that contains the OLAP Metadata Catalog where the metadata is stored
- One or more data sources that you want to use to create OLAP models and metaoutlines
- Essbase Integration Services

  For more information, see "Starting Integration Server" on page 14.
- Analytic Server

  For more information, see the *Essbase Integration Services Installation Guide*.

➤ To start Integration Services Shell, use one 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

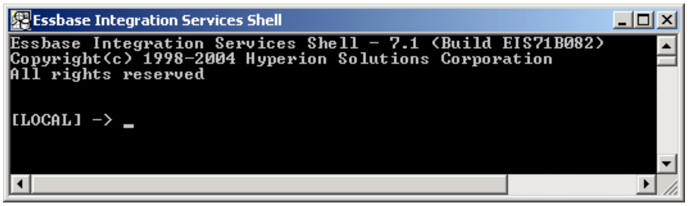
  ```
  olapicmd
  ```
- To set the TCP port number with which Essbase Integration Server communicates with its clients to a port different from the default 3388, enter the executable name and -P*portnumber* when you start Essbase Integration Services Shell; for example, type

  ```
  olapicmd -P3390
  ```

  Note: If 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 Integration Services Shell window opens, as shown in Figure 3:

Figure 3: Essbase Integration Services Shell Window



```
Essbase Integration Services Shell - 7.1 (Build EIS71B082)
Copyright(c) 1998-2004 Hyperion Solutions Corporation
All rights reserved

[LOCAL] -> _
```

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

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

About Integration Services Shell

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

With 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 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 108.

SET and LOAD Commands

SET commands tell Integration Server which data sources are involved in a load operation. LOAD commands tell 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 Integration Server with the location of the source databases, Analytic 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:

Command	Description
SETSOURCE	Specifies the data source databases from which an Analytic Services outline is built
SETTARGET	Specifies the name of the Analytic Server computer on which an outline for an Analytic Services application and database is built
SETCATALOG	Specifies the OLAP Metadata Catalog that you created to store OLAP models and metaoutlines

The STATUS command (see "STATUS" on page 111) 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:

Command	Description
LOADMEMBER	Loads members into an Analytic Services outline. You must load members before you load data.
LOADDATA	Loads data into an Analytic Services database. You cannot load data without loading members.
LOADALL	Loads both members and data into an Analytic Services 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 Integration Services Shell" on page 130 for a sample sequence of commands for loading members.

See "Loading Data Using Integration Services Shell" on page 131 for a sample sequence of commands for loading data.

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.

Command Syntax and Execution

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 data source 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), code page (CODEPAGE), 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;CODEPAGE=
English_UnitedStates.Latin1@Binary;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 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 Integration Services Console to schedule member or data loads, or to create load scripts. An 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. In the Schedule Essbase Load dialog box, be sure to check the Save Script Only option instead of the Schedule option. For more information, see "Recording Member and Data Load Scripts" on page 106.

Recording Member and Data Load Scripts

If you are creating scripts to perform loads or other tasks, you may want a script generated by 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 Integration Services by performing a load and recording it to a script file.

➤ To record a load script:

1. Start 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 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 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 platforms, a batch file (`.bat` extension) is an operating system file that calls multiple Integration Services Shell scripts and is used to run multiple sessions of 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, 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 Integration Services Shell:

```
olapicmd -fscriptFileName [> 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 Analytic Services from the Integration Server computer and generates outlines for a sample database. In the following example, the `status` command (see "STATUS" on page 111) is used to check the success or failure of each command. The `status` command returns the execution status in the Integration Services Shell window.

```
login labmachine1
status
settarget "DSN=labmachine1;UID=hyperion;PWD=password"
status
setcatalog "DSN=TBC_MD;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=tbc;PWD=password"
status
setsource "DSN=TBC;CODEPAGE=
   English_UnitedStates.Latin1@Binary;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
```

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 110
- “STATUS” on page 111
- “VERSION” on page 112
- “SETSOURCE” on page 112
- “SETTARGET” on page 113
- “SETCATALOG” on page 113
- “LOADMEMBER” on page 114
- “LOADDATA” on page 119
- “LOADALL” on page 122
- “LOGOUT” on page 128
- “SHUTDOWN” on page 129
- “EXIT” on page 129

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

The load commands in this topic are shown with an abbreviated syntax, which includes the syntax required for incremental loads. To see the full syntax for standard or incremental loads, use the Integration Services Console to record a script for a load. For more information on recording load scripts, see "Recording Member and Data Load Scripts" on page 106.

The following commands do not support aggregate storage:

- LOADMEMBER
- LOADDATA
- LOADALL

LOGIN

The LOGIN command connects you to the Integration Server computer. On a successful connection, the Integration Services Shell prompt changes to show the name of the Integration Server computer to which Integration Services Shell is connected. You can also use the LOGIN command to connect to an Integration Server computer that is running on a non-default port. Whether you are logging in to an Integration Server computer that is running on the default port or on a non-default port, this command returns an error if you already are connected to the Integration Server computer.

Logging in to Integration Server Running on the Default Port

Syntax

```
LOGIN IShost
```

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

Example

```
LOGIN cypress
```

Logging in to Integration Server Running on a Non-default Port

Syntax

```
LOGIN IShost:non-default_port_number
```

IShost	TCP/IP name or number of the computer where Integration Server is running
non-default_ port_number	The nonstandard port number on which Integration Server listens

Example

```
LOGIN cypress:3390
```

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 Integration Services Shell window:

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

VERSION

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

Syntax

```
VERSION
```

Example

VERSION returns the release number for Integration Services software.

SETSOURCE

The SETSOURCE command enables you to identify one or more source databases for Integration Server to use with a load command.

Syntax

```
SETSOURCE "DSN=PrimaryODBCdatasource;CODEPAGE=Codepage;
UID=username;PWD=password[;][DSN=SecondaryODBCdatasource;
CODEPAGE=Codepage;UID=username;PWD=password][;]...[DSN=
SecondaryODBCdatasource;CODEPAGE=Codepage;UID=username;
PWD=password]"
```

PrimaryODBC datasource	The primary ODBC data source name configured on the computer where Integration Server runs—case-sensitive
Codepage	The code page of the language you want to use during the current Integration Services Console session
username	The name for logging on to a primary or secondary ODBC data source—case-sensitive
password	The password for logging on to a primary or secondary ODBC data source—case-sensitive
SecondaryODBC datasource	The secondary ODBC data source name configured on the computer where Integration Server runs—case-sensitive

Example

```
SETSOURCE "DSN=TBC;CODEPAGE=English_UnitedStates.Latin1@Binary;
UID=TBC;PWD=Password;
DSN=TBC2;CODEPAGE=English_UnitedStates.Latin1@Binary;
UID=TBC;PWD=Password;DSN=TBC3;CODEPAGE=
English_UnitedStates.Latin1@Binary;UID=TBC;PWD=Password"
```

SETTARGET

The SETTARGET command enables you to identify a target Analytic Server computer for 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 Analytic Server is running
username	The name for logging on to Analytic Server
password	The password for logging on to Analytic 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 Integration Services to use with a load command.

Syntax

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

ODBCdatasource	The ODBC data source name configured on the computer where Integration Server runs—case-sensitive
Codepage	The code page of the language you want to use during the current Integration Services Console session

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;CODEPAGE=
English_UnitedStates.Latin1@Binary;UID=Fred;PWD=Password"
```

LOADMEMBER

The LOADMEMBER command initiates a member load operation on 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 "SET and LOAD Commands" on page 104.

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.

Note: The LOADMEMBER command does not support aggregate storage.

Syntax

```
LOADMEMBER "OTL=Metaoutline;APP=Essbase Application;
DBN=Essbase Database;[FLT_ID_MEMBER=Member_Load_Filter_ID;]
[DELETE_DB=Delete and Restore Database Y|N;]
[OTL_CLEAR=Delete All Members First Y|N;]
[INCUPD=DimID-MbrID,DimID-MbrID,...DimID-MbrID;]
[INCUPD_MEMBER=Dynamic Restructuring Member Load Options;]
[ATTR_UPDATE_LEVEL=Attribute Update Level;]
[@@USERS=Username List;][ESSCMD_SCRIPT=Esscmd Script File;]
[UNICODE=Create Unicode Application]
[UPDATE_DTDATA_ONLY=Update Drill-Through Data Only Y/N]"
```

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

Essbase Application The name of the Analytic Services application on the target Analytic Server computer where the Analytic Services outline is created.

Essbase Database

The name of the Analytic Services database on the target Analytic Server computer. Integration Server applies this name to the Analytic Services outline.

Member Load Filter ID

Optional. You can specify a member load filter ID to use when loading members. If you do not specify a filter ID, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. For information about creating filters, see the Integration Services Console Help.

Delete and Restore Database

Optional. The Delete and Restore Database parameter directs whether Integration Server should delete all members in the Essbase database before performing a member load. The default is N (No).

This is the format:

```
DELETE_DB=[Y|N]
```

When this option is selected (DELETE_DB=Y), the options Delete All Members First, Modify Essbase Users, and Incremental Update cannot be used.

Values are as follows:

Y Delete and restore (re-create) the Essbase database during the member load process.

N Do not delete and restore the Essbase database (this is the default).

Delete All Members First

Optional. When the Delete All Members First parameter is selected (OTL_CLEAR=Y), Integration Server removes all members from the existing database and then re-creates the database by using the members in the metaoutline.

This process is slower than creating or updating an Analytic Services 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 Analytic Services outline containing a smaller set of members. The default is N (No).

This is the format:

```
OTL_CLEAR=[Y|N]
```

Values are as follows:

Y Remove all members from the existing database and then re-create the database by using the members in the metaoutline.

N Do not remove members from the existing database; only update the Essbase database (this is the default).

DimID-MbrID, DimID-MbrID,... DimID-MbrID

Optional. Incremental Upate. The level at which dimensions are updated during an incremental member load. The format is:

```
INCUPD=DimID-MbrID,DimID-MbrID,
         ...DimID-MbrID;[;]
```

Note: If the data storage model property of the metaoutline is aggregate storage, this parameter cannot be used.

Values for incremental update are as follows:

- *DimID* represents the dimension ID of the dimension to which the incremental update applies.
- *MbrID* represents the member ID of the level zero member of the hierarchy to which the incremental update applies.

Dynamic Restructuring Member Load Options

Optional. Specify 1, 2, 3, or 4. Dynamic restructuring member load options set the parameters for restructuring the database during a member load. The option that you select affects any subsequent data loads.

Note: If the data storage model property of the metaoutline is aggregate storage, this parameter cannot be used.

Values are as follows:

1 Preserve all data. Select to preserve all existing data that applies to the changed outline when restructuring occurs. This is the default (`INCUPD_MEMBER=1`).

2 Discard all data. Select to clear all data from the database.

3 Preserve level 0 data. Select to preserve data only for level zero members.

This is the optimal restructure option if you change the source database and need to recalculate the data, and if all data required for the calculation is in level zero members.

Selecting this option deletes all upper-level blocks before restructuring. This reduces the disk space for restructuring and improves calculation time when the database is recalculated. The upper-level blocks are re-created when you calculate the database.

4 Preserve input data. Select to preserve only those blocks containing data that is loaded. Many applications contain data that is entered at parent levels. Selecting this option prevents deletion of any blocks that are created by data load, whether they are non-level zero or level zero (leaf member) blocks.

Attribute Update Level

Optional. Use with the Incremental Update parameter. The level at which attribute dimensions are updated when a member load is performed. Values are as follows:

0 Delete all attribute dimensions after the member load is performed.

1 Do not update existing attribute dimensions while updating other types of dimensions (for example: standard, time, or measures dimensions) during a member load.

2 Update all existing attribute dimensions and add new attribute dimensions during a member load.

Username List

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

Esscmd Script File

Optional. The name of an ESSCMD script. The script that you define instructs Analytic Services to perform specified actions on the Analytic Services 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 data source. The ESSCMD script must be in the *`ISHOME`*`/esscript` directory.

Create Unicode Application

Optional. You can specify that a Unicode application is created on Analytic Server. The default is that no Unicode application is created.

This parameter cannot be overwritten. For example, if a previous application exists and you are overwriting that application with the current member load, you cannot overwrite the original Unicode or non-Unicode setting.

This is the format:

```
UNICODE=[Y|N]
```

For example:

```
UNICODE=Y
```

Update Drill-Through Data Only

Optional. Update drill-through information or hybrid analysis information.

There is no separate option for hybrid analysis. Use this command for both drill-through information and hybrid analysis information.

This is the format:

```
UPDATE_DTDATA_ONLY=[Y|N]
```

For example:

```
UPDATE_DTDATA_ONLY=Y
```

Example

```
LOADMEMBER "OTL=ProductsAnalysis;APP=Products;
DBN=Analysis;FLT_ID_MEMBER=1;OTL_CLEAR=Y;
INCUPD=3-2,4-2,5-1,5-2,5-3;INCUPD_MEMBER=3;
@@USERS="TBC","CFO";UNICODE=Y;"
```

LOADDATA

The LOADDATA command initiates a data load operation on 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 on the SET commands, see “SET and LOAD Commands” on page 104.

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

Note: The LOADDATA command does not support aggregate storage.

Syntax

```
LOADDATA "OTL=Metaoutline;APP=Essbase Application;
DBN=Essbase Database;[FLT_ID_DATA=Data Load Filter ID;]
[REPLACE_ZEROS=Replace Zeros with #MISSING;]
[INCUPD=DimID-MbrID,DimID-MbrID,...DimID-MbrID;]
[INCUPD_DATA=Dynamic Restructuring Data Load Options;]
[@@USERS=Username List;][CALC_SCRIPT=Calc Script Name;]
[ESSCMD_SCRIPT=Esscmd Script File;]
[FT_COLUMN=FactTable Column for Incremental Update;]"
```

Metaoutline — The name of the metaoutline. Integration Server uses the specified metaoutline to extract data from the data source to create an Analytic Services outline.

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

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

Data Load Filter ID — **Optional.** You can specify a data load filter ID to use when loading data. If you do not specify a filter ID, the default filter (`*DEFAULT`) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. For information about creating filters, see the Integration Services Console Help.

Replace Zeros with #MISSING

Optional. The Replace Zeros with #MISSING parameter replaces Essbase account dimension values of zeros with #MISSING. The default is N (No).

This is the format:

```
REPLACE_ZEROS=[Y|N]
```

Here are the reasons that you may want to replace zeros with #MISSING:

- Fewer input blocks are created during the data load.
- The calculation time required in the Essbase database is decreased significantly.
- Because of differences in calculator behavior depending on whether a value is zero or missing, faulty data is not generated.

DimID-MbrID, DimID-MbrID,... DimID-MbrID

Optional. Incremental Update. The level at which dimensions are updated during an incremental data load. This is the format:

```
INCUPD=DimID-MbrID,DimID-MbrID,
        ...DimID-MbrID;[;]
```

Note: If the data storage model property of the metaoutline is aggregate storage, you cannot use this parameter.

Values for incremental update are as follows:

- *DimID* represents the dimension ID of the dimension to which the incremental update applies.
- *MbrID* represents the member ID of the level zero member of the hierarchy to which the incremental update applies.

Dynamic Restructuring Data Load Options

Optional. Dynamic restructuring data load options determine how Essbase loads values from a data source to the database. Specify 1, 2, or 3.

Note: If the data storage model property of the metaoutline is aggregate storage, you cannot use this parameter.

Values are as follows:

1 Overwrite. Select to replace the values in the database with the values in the data source. This is the default (INCUPD_DATA=1).

2 Add. Select to add values in the data source to the existing values in the database.

3 Subtract. Select to subtract the values in the data source from the existing values in the database.

Username List

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

Calc Script Name

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

Note: If the data storage model property of the metaoutline is aggregate storage, you cannot use this parameter.

Esscmd Script File

Optional. The name of an ESSCMD script. The script that you define instructs Analytic Server to perform specified actions on the Analytic Services 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 data source. The ESSCMD script must be in the *`ISHOME`*`/esscript` directory.

FactTable Column for Incremental Update

Optional. You can specify that a time-based incremental data load be performed based on the date the last data load was performed. The DateTimeColumn Name parameter is the name of the datetime column in the fact table that contains each record’s create date or modified date.

You can perform time-based incremental loads only if the fact table on which a metaoutline is based has a datetime column that contains each record's creation date or modified date.

This is the format:

```
FT_COLUMN=DataSourceName.TableName.ColumnName;
```

For example:

```
FT_COLUMN=TBC.Sales.Timestamp;
```

Example

```
LOADDATA "OTL=ProductsAnalysis;APP=Products;
DBN=Analysis;FLT_ID_DATA=2;REPLACE_ZEROS=Y;
CALC_SCRIPT=#DEFAULT#;INCUPD=1-2,2-1,3-2,4-2,5-1,5-2,5-3;
INCUPD_DATA=3;@@USERS="TBC","CFO";"
```

LOADALL

The LOADALL command initiates a load of both members and data on 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 on the SET commands, see “SET and LOAD Commands” on page 104.

Note: The LOADALL command does not support aggregate storage.

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 script to execute between a member load and a data load, use the LOADMEMBER and LOADDATA commands instead of the LOADALL command.

Syntax

```
LOADALL "OTL=Metaoutline;APP=Essbase Application;
DBN=Essbase Database;[FLT_ID_MEMBER=Member Load Filter ID;]
[FLT_ID_DATA=Data Load Filter ID;]
[DELETE_DB=Delete and Restore Database;]
[REPLACE_ZEROS=Replace Zeros with #MISSING;]
[OTL_CLEAR=Delete All Members First;]
[INCUPD=DimID-MbrID,DimID-MbrID,...DimID-MbrID;]
[INCUPD_MEMBER=Dynamic Restructuring Member Load Options;]
[INCUPD_DATA=Dynamic Restructuring Data Load Options;]
[ATTR_UPDATE_LEVEL=Attribute Update Level;]
[@@USERS=Username List;]
[CALC_SCRIPT=Calc Script Name;]
[ESSCMD_SCRIPT=Esscmd Script File;]
[UNICODE=Create Unicode Application]"
```

Metaoutline

The name of the metaoutline. Integration Server uses the specified metaoutline to extract data from the data source to create an Analytic Services outline.

EssbaseApplication

The name of the Analytic Services application on the target Analytic Server computer where the Analytic Services outline is created.

EssbaseDatabase

The name of the Analytic Services database on the target Analytic Server computer. Integration Server applies this name to the Analytic Services outline.

Member Load Filter ID

Optional. You can specify a member load filter ID to use when loading members. If you do not specify a filter ID, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. For information about creating filters, see the Integration Services Console Help.

Data Load Filter ID

Optional. You can specify a data load filter ID to use when loading data. If you do not specify a filter ID, the default filter (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. For information about creating filters, see the Integration Services Console Help.

Delete and Restore Database

Optional. The Delete and Restore Database parameter directs whether Integration Server should delete all members in the Analytic Services database before performing a member load. The default is N (No).

This is the format:

```
DELETE_DB=[Y|N]
```

When this option is selected (DELETE_DB=Y), the options Delete All Members First, Modify Essbase Users, and Incremental Update cannot be used.

Values are as follows:

Y Delete and restore (re-create) the Essbase database during the member load process.

N Do not delete and restore the Essbase database (this is the default).

Replace Zeros with #MISSING

Optional. The Replace Zeros with #MISSING parameter replaces Essbase account dimension values of zeros with #MISSING. The default is N (No).

This is the format:

```
REPLACE_ZEROS=[Y|N]
```

Here are the reasons that you may want to replace zeros with #MISSING:

- Fewer input blocks are created during the data load.
- The calculation time required in the Essbase database is decreased significantly.
- Because of differences in calculator behavior depending on whether a value is zero or missing, faulty data is not generated.

Delete All Members First

Optional. When the Delete All Members First parameter is selected (OTL_CLEAR=Y), Integration Server removes all members from the existing database and then re-creates the database by using the members in the metaoutline.

This process is slower than creating or updating an Analytic Services 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 Analytic Services outline containing a smaller set of members. The default is N (No).

This is the format:

```
OTL_CLEAR=[Y|N]
```

Values are as follows:

Y Remove all members from the existing database and then re-create the database by using the members in the metaoutline.

N Do not remove members from the existing database; only update the Essbase database (this is the default).

DimID-MbrID, DimID-MbrID,... DimID-MbrID

Optional. Incremental Update. The level at which dimensions are updated during an incremental member and data load. This is the format:

```
INCUPD=DimID-MbrID,DimID-MbrID,
         ...DimID-MbrID;[;]
```

Note: If the data storage model property of the metaoutline is aggregate storage, this parameter cannot be used.

Values for incremental update are as follows:

- *DimID* represents the dimension ID of the dimension to which the incremental update applies.
- *MbrID* represents the member ID of the level zero member of the hierarchy to which the incremental update applies.

Dynamic Restructuring Member Load Options

Optional. Specify 1, 2, 3, or 4. Dynamic restructuring member load options set the parameters for restructuring the database during a member load. The option that you select affects any subsequent data loads.

Note: If the data storage model property of the metaoutline is aggregate storage, this parameter cannot be used.

Values are as follows:

1 Preserve all data. Select to preserve all existing data that applies to the changed outline when restructuring occurs. This is the default (INCUPD_MEMBER=1).

2 Discard all data. Select to clear all data from the database.

3 Preserve level 0 data. Select to preserve data only for level zero members.

This is the optimal restructure option if you change the source database and need to recalculate the data, and if all data required for the calculation is in level zero members.

Selecting this option deletes all upper-level blocks before restructuring. This reduces the disk space for restructuring and improves calculation time when the database is recalculated. The upper-level blocks are re-created when you calculate the database.

4 Preserve input data. Select to preserve only those blocks containing data that is loaded. Many applications contain data that is entered at parent levels. Selecting this option prevents deletion of any blocks that are created by data load, whether they are non-level zero or level zero (leaf member) blocks.

Dynamic Restructuring Data Load Options

Optional. Dynamic restructuring data load options determine how Essbase loads values from a data source to the database. Specify 1, 2, or 3. This is the default (INCUPD_DATA=1).

Note: If the data storage model property of the metaoutline is aggregate storage, this parameter cannot be used.

Values are as follows:

1 Overwrite. Select to replace the values in the database with the values in the data source.

2 Add. Select to add values in the data source to the existing values in the database.

3 Subtract. Select to subtract the values in the data source from the existing values in the database.

Attribute Update Level

Optional. Use with the Incremental Update parameter. The level at which attribute dimensions are updated when a member load is performed. Values are as follows:

0 Delete all attribute dimensions after the member load is performed.

1 Do not update existing attribute dimensions while updating other types of dimensions (for example: standard, time, or measures dimensions) during a member load.

2 Update all existing attribute dimensions and add new attribute dimensions during a member load.

EssbaseUsers

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

Calc Script Name

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

Note: If the data storage model property of the metaoutline is aggregate storage, this parameter cannot be used.

Esscmd Script File

Optional. The name of an ESSCMD script. The script that you define instructs Analytic Server to perform specified actions on the Analytic Services 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 data source. The ESSCMD script must be in the `*ISHOME*\esscript` directory.

Create Unicode Application

Optional. You can specify that a Unicode application is created on Analytic Server. The default is that no Unicode application is created.

This parameter cannot be overwritten. For example, if a previous application exists and you are overwriting that application with the current member load, you cannot overwrite the original Unicode or non-Unicode setting.

This is the format:

```
UNICODE=[Y|N]
```

For example:

```
UNICODE=Y
```

Example

```
LOADALL OTL=ProductsAnalysis;APP=Products;
DBN=Analysis;FLT_ID_MEMBER=2;FLT_ID_DATA=2;
REPLACE_ZEROS=Y;OTL_CLEAR=Y;CALC_SCRIPT=#DEFAULT#;
INCUPD=1-2,2-1,3-2,4-2,5-1,5-2,5-3;INCUPD_MEMBER=3;
INCUPD_DATA=3;@@USERS="TBC","CFO";UNICODE=Y;
```

LOGOUT

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

Syntax

```
LOGOUT
```

Example

LOGOUT logs you out of Integration Server.

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

SHUTDOWN

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

Syntax

```
SHUTDOWN
```

Example

SHUTDOWN shuts down Integration Server.

EXIT

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

Syntax

```
EXIT
```

Example

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

Loading Dimensions and Members Using Integration Services Shell

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

You can manually schedule the batch file by using the AT service in Windows 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 Integration Services Shell, follow the same basic steps for loading members as you would when using Integration Services Console:

1. Log on to the 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;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password
   ```

3. Connect to the Analytic 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;CODEPAGE=
   English_UnitedStates.Latin1@Binary;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 Integration Services Shell

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

You can manually schedule the batch file by enabling the Task Scheduler service on Windows 2000 and Windows XP 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 Integration Services Shell, follow the same basic steps for loading data as you would using Integration Services Console:

1. Log on to 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;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password
   ```

3. Connect to the Analytic 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;CODEPAGE=
   English_UnitedStates.Latin1@Binary;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"
   ```

Appendix A Return Codes and Error Messages

This appendix provides a partial list of the return codes and error messages generated by Essbase Integration Server and Essbase Analytic Server. Return codes and error messages are grouped into two categories: return codes and errors generated by Integration Server and errors generated by Analytic Server for member and data loads.

The following sections contains return codes and errors generated by Integration Server:

The complete lists of OLAPICMD return codes (`unixrtcd.txt` and `winrtcd.txt`) are located in `$ISHOME/docs` on UNIX and `ISHOME\docs` on Windows.

The complete list of Integration Server error messages is located in `$ISHOME/bin/error.txt` on UNIX and `ISHOME\bin\error.txt` on Windows.

For errors generated by Analytic Services for member and data loads, see "Essbase Error Messages Generated During Data Loads" on page 143.

OLAPICMD Session Return Codes

At the end of an Integration Services Shell (OLAPICMD) session, OLAPICMD may return a code generated by Integration Server. This return code may be a status code or may be an error code.

To interpret the meaning of a return code, refer to the `unixrtcd.txt` file or the `winrtcd.txt` file. The `unixrtcd.txt` file is located in the `$ISHOME/docs` directory on UNIX, and the `winrtcd.txt` file is located in the `ISHOME\docs` directory on Windows.

- The `unixrtcd.txt` file lists the return codes for UNIX, the corresponding Windows return code, and the associated message text. Because of a limitation in the number of return code values available for use on UNIX, some code numbers are used more than once. When you look up a return code number and find that it is used multiple times, you need to examine the messages for all instances of that return code number. Then you need to determine which message applies to your situation by understanding the context in which the problem occurred.
- The `winrtcd.txt` file lists the return codes for Windows and the associated message text. The return code values in Windows are unique.

Table 8 shows examples of return codes on UNIX that are used more than once. Italicized words represent variable names inserted in the message text. Refer to the return code value and make a note of it in case you need to contact Hyperion Technical Support with a problem.

For the complete list of return codes, refer to the `unixrtcd.txt` (located in the `$ISHOME/docs` directory on UNIX) and the `winrtcd.txt` file (located in the `ISHOME\docs` directory on Windows).

Table 8: Examples of Multiple Instances of Integration Services Shell (OLAPICMD) Return Code Values for UNIX

UNIX Return Code	Windows Return Code	Message Contents
061	1195069	Failed to add Essbase member *member_name* (DUPLICATE, Ignored).
061	1195325	There must be an active session to process this command.

Table 8: Examples of Multiple Instances of Integration Services Shell (OLAPICMD) Return Code Values for UNIX(Continued)

UNIX Return Code	Windows Return Code	Message Contents
061	2003005	Data source error. Could not locate the metadata attributes.
069	1195077	Failed to create a local context.
069	1195333	ODBC Error. Encountered unknown ODBC exception while opening database.
094	1195358	Failed to get user attributes.
094	2002014	Cube builder error. Client request error. Invalid number of parameters.

Member Load Error Messages

Table 9 lists some of the error messages that 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 9: Essbase Integration Server Member Load Error Messages

Windows Message Number	Message Content
1195049	Detected metadata join problem (*table.column* < - > *table.column*).
1195050	Encountered unknown ODBC exception while opening database. Aborting the command.
1195064	Failed to add database *Essbase database*.
1195065	Failed to add dimension *dimension*.
1195066	Failed to add Essbase member *member* (%d).
1195067	Failed to add Essbase member *member* (child of shared parent).
1195068	Failed to add Essbase member *member* (DUPLICATE).

Table 9: Essbase Integration Server Member Load Error Messages (Continued)

Windows Message Number	Message Content
1195069	Failed to add Essbase member *member* (DUPLICATE, ignored).
1195070	Failed to add Essbase member *member* (no parent or sibling members found).
1195071	Failed to add Essbase member *member*, %d.
1195072	Failed to add Essbase member *member;* the member exists.
1195073	Failed to add formula *formula*.
1195074	Failed to add member *member.*
1195075	Failed to add metaoutline member *member.*
1195076	Failed to add user attribute *user-defined attribute*.
1195077	Failed to create a local context.
1195078	Failed to create a local outline.
1195079	Failed to create application *Essbase application*.
1195080	Failed to create dimension *dimension*.
1195083	Failed to get information for parent of *recursive hierarchy member.*
1195085	Failed to get metadata for *dimension.member.*
1195088	Failed to initialize Essbase API.
1195089	Failed to open outline *Essbase outline*.
1195093	Failed to restructure *Essbase outline*.
1195095	Failed to update Essbase server (NULL HEADER).
1195097	Filter *metaoutline filter* doesn't exist for *metaoutline* metaoutline.
1195098	Filter expression length exceeded the limit of *maximum bytes* bytes.
1195115	Incremental update specification *incremental update specification string* is incorrect.
1195126	Invalid aggregation function *aggregate function* for *table.column*.
1195129	Invalid esscmd script name *esscmd script*.

Table 9: Essbase Integration Server Member Load Error Messages (Continued)

Windows Message Number	Message Content
1195130	Invalid parameter *parameter*.
1195132	Invalid parameter string.
1195142	Metadata error: *property ID* is not a valid property for *member* or *dimension*.
1195143	Metaoutline validation error. *Recursive hierarchy member*, a recursive member cannot have aggregation filter.
1195144	Metaoutline validation error. Filter = *member filter*, is not associated with any member.
1195188	Unknown exception encountered while opening dimension *dimension*.
1195189	Unknown exception encountered while opening member *member*.
1195190	Unknown exception encountered while opening metaoutline *metaoutline*.
1195341	Failed to Login to Essbase Server.
1195342	Failed to Get Essbase Settings.
1195344	Failed to unload database.
1195345	Failed to delete database.
1195346	Failed to close outline.
1195347	Failed to read outline.
1195349	Failed to unlock Database *database*.
1195350	Failed to save outline.
1195351	Unable to process state for *Essbase command*.
1195352	Failed to get access permissions for application.
1195353	Failed to set access permissions for application.
1195354	Failed to get access permissions for database.
1195355	Failed to set access permissions for database.

Table 9: Essbase Integration Server Member Load Error Messages (Continued)

Windows Message Number	**Message Content**
1195356	No member found.
1195357	Essbase Error: *error definition*.
1195358	Failed to get user attributes.
1195360	Failed to add member to outline.
1195361	Failed to calculate the data.
1195362	Failed to get parent.
1195363	Failed to get member.
1195364	Failed to move member.
1195365	Failed to get shared member.
1195366	Failed to get member information.
1195367	Failed to sort outline.
1195377	IS Error: Unable to delete outline.
1195378	IS Error: SQL Generation for member load failed.
1195379	IS Error: Database close operation failed after load.
1195380	IS Error: Catalog close operation failed after load.
1195381	IS Error: Essbase close operation failed after load.
1195382	IS Error: No OLAP model name specified.
1195383	IS Error: Failed to open OLAP model *OLAP model*.
1195384	IS Error: Failed to close OLAP Metadata Catalog *OLAP Metadata Catalog*.
1195385	Failed to create virtual model.
1195386	Member load completed with errors.
1195387	Member load terminated due to error.
1195389	Failed to get dimbuild SQL.

Data Load Error Messages

Table 10 lists some of the error messages that 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 10: Essbase Integration Server Data Load Error Messages

Windows Message Number	Message Content
1195034	Cell load exceptions encountered.
1195047	Data load exceptions encountered. *?* , *amount*.
1195063	Failed to activate *Essbase_application.Essbase.database*.
1195082	Failed to execute esscmd script *esscmd script*.
1195369	Data load completed with errors.
1195370	Data load terminated with errors.
1195388	Failed to get data load SQL.
1195390	Number of data load SQLs not equal to number of hierarchies.

Drill-Through Report Error Messages

Drill-through reports are created using the Integration Services Console OLAP Metaoutline standard user interface and are viewed using Essbase Spreadsheet Add-in for Excel or Lotus 1-2-3 or any other Hyperion drill-through client.

Table 11 lists some of the error messages that Integration Server may generate to Analytic Services 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 11: Essbase Integration Server Drill-Through Report Error Messages

Windows Message Number	**Message Content**
1195081	Failed to enable DTS Member *member*.
1195087	Failed to get the universal member handle.
1195125	Intersection element *member* does not exist in *table.column*.
1195131	Invalid parameter count for the drill-through request. Aborting...
1195153	Received NULL pointer in DTAttributes( ) for Report = *Drill-Through report*.
1195154	Received NULL pointer in GetDTData ( ) for Report = *Drill-Through report*.
1195155	Received NULL pointer in GetDTDomain ( ).
1195156	Received NULL pointer in GetDTReport ( ).
1195157	Received NULL pointer in SetDTReport ( ).
1195183	Too many Drill-Through reports defined (exceeded Essbase metadata size limit). Drill-Through is disabled for this outline.
1195186	Unknown exception in GetDTReport ( ).
1195359	Essbase Error: Invalid Drill-Through Metadata.
1195369	IS Error: Data load completed with errors.
1195370	IS Error: Data load terminated due to errors.
1195371	IS Error: Unable to log in to data source.
1195372	IS Error: Unable to log in to OLAP Metadata Catalog.
1195373	IS Error: Unable to log in to Essbase Server.
1195374	IS Error: Unable to read Metaoutline information.

Table 11: Essbase Integration Server Drill-Through Report Error Messages (Continued)

Windows Message Number	Message Content
1195375	IS Error: Data calculation failed.
1195376	IS Error: Esscmd script execution failed.

Miscellaneous Error Messages

Table 12 lists some of the miscellaneous error messages that 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 12: Essbase Integration Server Miscellaneous Error Messages

Windows Message Number	Message Contents
1195004	*Metaoutline* failed validation. Database measures not specified.
1195007	*Add job* failed.
1195017	*Syntax* Syntax error at *character location* in the filter expression *filter*.
1195018	*Member* cannot have aggregation filter. Only leaf members can have aggregation filters.
1195054	Error message unavailable for this error.
1195057	Essbase server on *Essbase server computer* is not enabled with Essbase Integration Services option. Please inform your system administrator.
1195084	Failed to get job info (OS error *Job-ID*).
1195086	Failed to get the ODBC message.
1195090	Failed to remove *temporary batch file* (OS error *operating system error*).

Table 12: Essbase Integration Server Miscellaneous Error Messages (Continued)

Windows Message Number	Message Contents
1195091	Failed to remove job (OS error *operating system error*).
1195092	Failed to remove old job (OS error *operating system error*).
1195094	Failed to schedule job (OS error *operating system error*).
1195096	Fatal error.
1195124	Internal system error. Please contact Technical Support with the error number 1999999.
1195127	Invalid command line option *OLAP Integration Server switch*.
1195128	Invalid datatype specification *data type*.
1195133	Invalid session identifier; please log in to Essbase Integration Services again.
1195159	Replace Failed *member transformation.*
1195185	Trying to access invalid memory. Contact Technical Support.
1195187	Unknown ODBC exception encountered while closing database.
1195206	*Syntax* Syntax error at *character location* in the filter expression *filter.*
1195207	Essbase Integration Services is already running on this machine.
1195208	Failed to release locks.
1195333	ODBC Error: Encountered unknown ODBC exception while opening database.
1195334	ODBC Error: Encountered unknown ODBC exception while closing database.
1195336	ODBC Error: Failed to get ODBC Error message for error number.
1195338	IS Error: Error during parsing Load Parameters.
1195343	Essbase Error: Essbase Server on *Essbase server computer* is not enabled with Essbase Integration Services option.
1195344	Failed to unload database.
1195348	Failed to get information for Database *database*.

Table 12: Essbase Integration Server Miscellaneous Error Messages (Continued)

Windows Message Number	Message Contents
1195391	IS Error: Unable to get tables for pattern name *SQL regular expression pattern* from data source.
1195392	IS Error: Unable to get columns for table *data source table* from data source.
1195393	IS Error: Unable to extract foreign keys for tables *data source table, data source table* from data source.
1195394	IS Error: Failed to bind columns.
1195395	IS Error: Failed to fetch next row.
1195396	IS Error: Failed to get number of columns.
1195397	IS Error: Failed to get column attributes.
1195398	IS Error: Failed to get statement handle.
1195399	IS Error: Failed to release statement handle.
1195400	IS Error: Failed to get number of rows.
1195401	IS Error: Unable to delete the model *model name*.

Essbase Error Messages Generated During Data Loads

Table 13 lists the most common errors that you will find in the `dataload.txt` file. The `dataload.txt` file for a specific data load is located in a subfolder in the `ISHOME\loadinfo` directory in Windows and `$ISHOME/loadinfo` in UNIX. In the `loadinfo` directory, each data load generates a subfolder named in the following format:

```
<application_database_timestamp_sessionnumber>
```

Locate the subfolder for the data load that you want to review and open the `dataload.txt` file to view the errors.

Table 13: Essbase Messages Generated During Loads

Message Number	Message Contents
3303	Member not found in database.
3304	Insufficient access to store data.
3333	Bad data value supplied.
3335	Record rejected because of duplicate member names.
3336	Member/Data unknown.
3337	Record rejected because of dimension conflicts with Header Name.

Appendix

B

Integration Services Limits and Guidelines

This appendix contains the following sections related to limits and guidelines for Essbase Integration Services:

For limits on naming OLAP model and metaoutline objects, see the naming conventions topics in the Essbase Integration Services Console online help.

Object Limits

Table 14 contains a list of limits that you may encounter when creating or manipulating Integration Services objects.

Table 14: Objects and Limits

Object	Limit for Applications
Names and Related Fields	
Alias name	Non-Unicode application limit: 80 bytes Unicode-mode application limit: 80 characters

Table 14: Objects and Limits (Continued)

Object	Limit for Applications
Alias table name	Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters
Analytic Server name	Non-Unicode application limit: 29 bytes Unicode-mode application limit: 50 characters
Application name	Non-Unicode application limit: 8 bytes Unicode-mode application limit: 30 characters
Application description	Non-Unicode application limit: 79 bytes Unicode-mode application limit: 80 characters
• Custom-defined function name • Custom-defined macro name • Custom-defined function specification • Custom-defined macro specification	Non-Unicode application limit: 127 bytes. MaxL and the API truncate characters after 127 bytes. Unicode-mode application limit: 128 characters. MaxL and the API truncate characters after 128 characters. In either case, no truncation on server. No error is displayed if truncation occurs.
Custom-defined function and macro comment	Non-Unicode application limit: 255 bytes. After 255 bytes, characters are truncated by MaxL and API. Unicode-mode application limit: 256 characters. MaxL and the API truncate characters after 256 characters. In either case, no truncation on server. No error is displayed if truncation occurs.
Data source name	46 characters
Database name	Non-Unicode application limit: 8 bytes Unicode-mode application limit: 30 characters
Database description	Non-Unicode application limit: 79 bytes Unicode-mode application limit: 80 characters
Directory path For example: `/essintegration/bin`	Non-Unicode application limit: 256 bytes Unicode-mode application limit: 1024 bytes

Table 14: Objects and Limits (Continued)

Object	Limit for Applications
File names for calculation scripts, report scripts, and rules files	Non-Unicode application limit: 8 bytes Unicode-mode application limit: If included within a path, the smaller of the following two values: • 1024 bytes • The limit established by the operating system If not included within a path, as in some MaxL statements, 1024 bytes.
Filter name	Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters
Group name	Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters
Linked reporting object cell note	599 bytes
Linked reporting object URL	512 characters (always single-byte characters)
Member comment field	Non-Unicode application limit: 255 bytes Unicode-mode application limit: 256 characters
Member comment field (extended)	8192 bytes
Member name	Non-Unicode application limit: 80 bytes Unicode-mode application limit: 80 characters
OLAP model name	80 characters
OLAP metaoutline name	80 characters
Password	Non-Unicode application limit: 100 bytes Unicode-mode application limit: 100 characters
Substitution variable name	80 bytes
Substitution variable value	255 bytes
Trigger name	30 bytes
User-defined query	No limitations
User names	Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters

Table 14: Objects and Limits (Continued)

Object	Limit for Applications
Variable names	32 bytes
Data Load and Dimension Building Limits	
Data load query	No limitations
Member load query	No limitations
Number of alias tables associated with a metaoutline	9, including the Default alias table
Number of error messages written to a data load or dimension build error log (DATAERRORLIMIT in `essbase.cfg`)	Default 1000, minimum 1, maximum 65000
Selection and rejection criteria	Number of characters that describe selection and rejection criteria: combination of all criteria limited to 32 KB
Other Limits	
Caches: data, data file, index	2 GB
Formula size	• Created in Formula Editor: 64 KB. Formulas in calculation scripts are not subject to this limit. • Created in MaxL, using multi-byte characters: 40 KB. Formulas in calculation scripts are not subject to these limits.
Number of members in a metaoutline	1,000,000 explicitly defined in an Essbase outline. Because Hybrid Analysis and some uses of partitions can involve many more members than are explicitly listed in an outline, the actual number of members accessible through the database is much higher.
Number of security filters	Per Analytic Server, 65535 Per Analytic Services database, 32290
Number of users	30,000. Errors can occur if you create more than 30,000 users.

Table 14: Objects and Limits (Continued)

Object	Limit for Applications
Number of members in an Analytic Services outline	Approximately 1,000,000 explicitly defined in an Analytic Services outline. Hybrid Analysis and some uses of partitions enable access to many more members than are explicitly listed in an outline, the actual number of members accessible through the database is much higher. Longer names, which often occur if multi-byte characters are used, decrease the number of members that are allowed.

Source Database Object Naming Limits

Integration Services does not support source database table names and column names that contain the characters listed in Table 15.

Note: The database object naming limits specified in Table 16 apply to SAP BW Info Objects (QueryCubes and InfoCubes).

Table 15: Source Database Object Naming Limits for Table Names and Column Names

	Character Description		Character Description
"	quotation mark	( )	parentheses
&	ampersand	.	period
,	comma	|	vertical pipe
=	equal sign	'	single quotation mark
@	at sign	\	backslash
#	pound sign	/	forward slash
\t	Tab character	:	colon
	blank space	*	asterisk
+	plus sign	?	question mark

Table 15: Source Database Object Naming Limits for Table Names and Column Names (Continued)

	Character Description		Character Description
-	dash, minus sign, or hyphen	<	less than sign
{ }	braces	>	more than sign

Hybrid Analysis Guidelines

You should be familiar with the Hybrid Analysis guidelines covered in this section.

Data Source Guidelines

- A single Analytic Services database can be associated with only one hybrid analysis data source.
- A hybrid analysis data source can consist of only one data source database.
- Hybrid Analysis does not support SAP BW data warehouses.
- Hybrid Analysis supports data that is stored using either block storage or aggregate storage.
- Hybrid Analysis supports Unicode-enabled data sources.

Dimensions Guidelines

- Hybrid Analysis is not supported on accounts dimensions.
- If the time dimension contains hybrid analysis-enabled members, the time dimension does not support Dynamic Time Series.
- Hybrid Analysis is not supported on user-defined dimensions.
- You can perform operations and analyses on dimensions that have attributes attached to one or more levels in an outline that is hybrid analysis-enabled. The attribute dimension should be fully loaded into Analytic Services.
- Only the first hierarchy of a dimension with alternate hierarchies can have members enabled for hybrid analysis on its lowest levels.
- When building a dimension that is enabled for hybrid analysis, you must ensure that the column in the data source table that contributes to the leaf level of the Analytic Services portion of the dimension is non-nullable.

Members Guidelines

- Only the lowest level members of a dimension can be enabled for hybrid analysis.
- You should not rename a hybrid analysis-enabled member. If you rename a member, the member may not be retrieved the next time you perform a drill-through operation.
- Hybrid Analysis supports only parent-child prefixing on member names.
- Analytic Services does not support aliases for members that are enabled for hybrid analysis.
- Hybrid Analysis does not support scaling of measures dimension members using any of the operators + (addition), - (subtraction), * (multiplication), and / (division). If you use the scaling operators, drill-through queries into hybrid analysis data may show a mismatch between aggregated level-0 values in the Analytic Services database and the corresponding detail values in your data source.
- Analytic Services ignores all member properties, such as formulas, UDAs, and aliases for members that are enabled for hybrid analysis.
- You can associate an attribute member with a member enabled for hybrid analysis but you must make the association by metaoutline member name and not by level.
- Analytic Services supports drill-through operations defined on members that are enabled for Hybrid Analysis.

Spreadsheet Guidelines

- Hybrid Analysis does not return numeric data in a spreadsheet if the member from the Accounts dimension is part of a ragged hierarchy.
- Hybrid Analysis is not supported with the Member Selection feature. You cannot select Hybrid Analysis members from the Member Selection dialog box.
- If you have multiple levels of hybrid analysis members in your outline, performing a zoom out operation on the bottom level hybrid analysis member takes you directly to the Essbase parent member, bypassing all other hybrid analysis levels.

Operations Guidelines

- Hybrid Analysis supports Dynamic Time Series.
- If the time dimension contains hybrid analysis-enabled members, the time dimension does not support Dynamic Time Series.
- Analytic Services requires the OLAP Metadata Catalog created in Integration Services in order to drill down in a hybrid analysis data source.
- Hybrid Analysis does not support transparent, replicated, or linked partitions.
- Hybrid Analysis does not support recursive hierarchies.

Limitations on Transformation Rules with Hybrid Analysis

Hybrid Analysis sets the following limitations on transformation rules:

- A database value cannot have a separator character that is the same as the one used for the prefix or suffix.
- A member name cannot be more than 80 characters (excluding blanks).
- A prefix or suffix must always have a separator associated with it.
- The data source database value cannot have trailing blanks.
- If spaces are converted to underscores during a transformation, then the Hybrid Analysis Manager assumes there are no underscores present in the database value.
- The use of all ancestors as a prefix or as a suffix is not supported.

Transformations Not Supported by Hybrid Analysis

Hybrid Analysis does not support the following transformations:

- Dropping spaces from around a member name
- Applying a prefix without a separator
- Applying names prefixes for all parents with or without a separator
- Applying a suffix without a separator
- Applying all suffixes of parent names with or without a separator
- Applying scaling to measures

Limitations on Using Formulas with Hybrid Analysis

Formulas used with hybrid analysis-enabled members are subject to the following limitations:

- Formulas are supported only on a measures dimension.
- Formulas cannot be attached to relational members.
- Formulas cannot reference a relational member by name.
- Member set functions (such as @CHILDREN and @DESCENDANTS), which generate member lists in a formula, execute only in the Analytic Services portion of the outline.

If a formula contains one or more functions that are not supported by Hybrid Analysis, Analytic Services returns the following error message:

```
Error executing formula for member
[member-name-to-which-formula-is-attached] (line [line# where
the offending function appears inside the formula):
function [Name of the offending function] cannot be used in
Hybrid Analysis.
```

Unsupported Analytic Services Functions in Hybrid Analysis

Hybrid Analysis does not support all Analytic Services functions. The following topics specify the categories of significant Analytic Services functions not supported by Hybrid Analysis.

Relationship Functions

Hybrid Analysis does not support functions that look up specific values in the database based on current cell location and a series of parameters. Some examples of these functions are given next.

@ANCEST
@SANCEST
@PARENT
@SPARENT
@CURLEV
@CURGEN

Specifying Member Conditions Using Boolean Test

Hybrid Analysis does not support functions used to specify member conditions. Some examples of these functions are listed next.

@ISIANCEST
@ISIPARENT
@ISISIBLING
@ISLEV
@ISSAMEGEN
@ISUDA

Range Functions

Hybrid Analysis does not support functions that take a range of members as arguments. Rather than return a single value, these functions calculate a series of values internally based on the range specified. Some examples of range functions that are not supported are listed next.

@PRIOR
@SHIFT
@PRIORS
@SHIFTS
@NEXT
@MDSHIFT
@MOVSUM
@MOVAVG
@ALLOCATE
@MDALLOCATE
@VAR
@VARPER
@MEDIAN
@RANK

Attribute Functions

Hybrid Analysis does not support any Essbase functions that deal with attributes. Some examples of these functions are listed next.

@ATTRIBUTEVAL
@WITHATTR
@ATTRIBUTESVAL

Current Member and XREF

Hybrid Analysis does not support the following functions used to determine whether the current member is the member being specified.

@CURRMBR
@XREF

Drill-Through Report Guidelines

You should be familiar with these drill-through report guidelines:

- Drill-through operations can be performed on a Unicode Analytic Services database.
- Drill-through operations are not supported in SAP BW data warehouses.
- Drill-through operations can be performed with Dynamic Time Series.
- If you are creating drill-through reports for a metaoutline that is based on multiple data sources, the SQL used for each drill-through report cannot include a join across a data source boundary; that is, each drill-through report can be based on only one data source. A metaoutline can contain multiple drill-through reports; each, however, must be based on a single data source.
- Multi-cell drill-through is supported under the following conditions:
  - All the members selected for multi-cell drill-through should come from same physical table and column in the database.
  - All the members selected for multi-cell drill-through should come from the same level in the metaoutline.
  - The selected members cannot come from more than one hierarchy.
- The metaoutline in which you are designing a drill-through report cannot contain alternate recursive hierarchies. For example, you cannot insert the same member from a recursive hierarchy into a dimension more than once.
- You cannot use the $$ substitution variable in the template SQL (user-defined SQL) for a drill-through report when the intersection level of the dimension is defined at Generation 1 and the dimension is built from a parent/child table.

Unicode Guidelines

- You must use the manual catalog creation procedures to create a Unicode-enabled OLAP Metadata Catalog, as described in *Essbase Integration Services Installation Guide*.
- Drill-through operations can be performed on a Unicode Analytic Services database.
- Hybrid Analysis does not support Unicode-enabled databases.

- Integration Services does not support Unicode databases in SQL Server, Sybase and Informix RDBMSs.
- If the database is UTF-8 but you are using only one language, select the UTF-8 option from the Code Page drop-down list whenever you access the database with the Login, OLAP Metadata Catalog Setup, Set Login Defaults, and Data Source dialog boxes.
- Unicode is supported in SAP BW data sources. If the data in your SAP BW data source is not in English, select the UTF-8 option from the Code Page drop-down list whenever you access the database.

DB2 Cube Views Restrictions

You can use a utility called a *bridge* to import OLAP model and metaoutline metadata objects from DB2 Cube Views into Integration Services. You use the bridge to import your models or metaoutlines to an XML file. The XML file maps Integration Services metadata objects with DB2 Cube Views metadata objects. The XML file is then used to import the metadata objects from the DB2 catalog into Integration Services metadata catalog.

There are several restrictions on mapping metadata objects from DB2 Cube Views to Integration Services:

- Cube Views hierarchies that have recursive deployment are mapped to recursive hierarchies in Integration Services.

  All hierarchy types (balanced, unbalanced, ragged, and network) that have standard deployment in Cube Views are mapped to regular hierarchies in Integration Services.
- Unsupported aggregation functions from CubeViews are mapped to "None" in Integration Services.
- Members composed in CubeViews are translated into model view members in Integration Services in the following manner:

  ```
  physicalTable="CubeView(NoTable)"
  physicalColumn="CubeViewExpr(Templatefor CubeView member)"
  ```

  and

  ```
  extractionRule="CubeViewParam($$1=
  CubeView-membername-$$2=CubeView-member-name-2 ...)"
  ```

Elements with a CubeView prefix need to be mapped and defined by the Integration Services/CubeView administrator as appropriate.

- Model logical join members may not match corresponding view names.

  The XML model logical join members may not match their corresponding view names and could be swapped. This may occur when there are complex multiple or recursive joins around Facts in CubeView. As a workaround, swap or delete the join member information from the generated XML model file.

- Joins are not created for dimensions.

  In generated OLAP models, joins are not created for dimensions with snowflake schemas. As a workaround, you can connect the dimension views in the Integration Services Console.

- A CubeView XML must contain at least one CubeModel.
- A CubeView XML cannot contain multiple cubes. It may contain "None" or "One" cube.

Note: For information on the use of DB2 Cube Views, see your IBM DB2 Cube Views product documentation.

Unsupported Aggregate Functions in SAP BW

The following aggregate functions used in SAP BW data warehouses are not supported in Integration Services:

- AGGR_UNKNOWN
- AGGR_VAR (variance)
- AGGR_STD (standard deviation)
- AGGR_CALCULATED

The aggregate function is defined at the measures dimension. If a measures dimension contains an aggregate function not supported in Integration Services, the aggregate function is mapped to <NONE>.

Glossary

accounts dimension. A dimension type that makes accounting intelligence available. Only one dimension can be tagged as Accounts. An accounts dimension is not necessary.

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

Administration Server. A Java middle-tier server that handles communication between the Administration Services Console and Essbase Analytic Servers. It authenticates users as they connect to Analytic Servers.

Administration Services. An Essbase product used to manage the Essbase environment from a single point of access. It consists of a Java client console and middle-tier server that communicate directly with Essbase Analytic Servers.

Administration Services Console. A Java graphical user interface that communicates with Essbase Analytic Servers by way of the Essbase Administration Server. The console enables administrators to manage Analytic Servers from a single, graphical representation of the Essbase environment.

aggregate. *See* consolidate.

aggregate cell. In aggregate storage databases, cells with values that are rolled up from lower-level cell values and potentially stored in aggregations. Input (level 0) cells and accounts dimension cells are not aggregate cells.

aggregate storage database. The Essbase database storage model supporting large-scale, sparsely distributed data, which is categorized into many, potentially large dimensions. Selected data values are aggregated and stored, typically with improvements in aggregation time.

aggregate view. A collection of aggregate cells based on the levels of the members within each dimension. For example, a single aggregate view in a three-dimension outline can be represented as 0,2,1. The numbers are listed in the order of the dimensions in the outline. Each number represents the member level in the dimension. If the dimensions are Measures, Products, and Geography, the aggregate view 0,2,1 includes aggregate cells for intersections including level 0 members of Measures, level 2 members of Products, and level 1 members of Geography. Aggregate views are not defined for accounts dimension levels above level 0.

aggregation. The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

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.

alias table. A database table that stores aliases for the dimensions or members.

alternate name. *See* alias.

Analytic Server. The server component of Analytic Services.

Analytic Server log. A record of actions performed by the Analytic Server (agent).

Analytic Services database. A repository of data within Essbase Analytic Services 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 calc scripts or data load rules. *See also* application.

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 Essbase Analytic Services, a container for one or more Analytic Services databases and files, such as calc 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. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

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

block storage database. The Essbase database storage model categorizing and storing data based on the sparsity of data values defined in sparse dimensions. Data values are stored in blocks, which exist only for sparse dimension members for which there are values.

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

Builder. *See* Essbase Integration Services Console.

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.

child. A member that has a parent above it in the database outline.

code page. A mapping of bit combinations to a set of text characters. Different code pages support different sets of characters. Each computer contains a code page setting for the character set requirements of the language of the computer user. In the context of this document, code pages map characters to bit combinations for non-Unicode encodings. *See also* encoding.

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.

currency. The monetary unit of measure associated with a balance or transaction.

currency conversion. A process that converts currency values in a database from one currency into another currency. For example, to convert one U. S. dollar into the euro, the exchange rate (for example, 0.923702) is multiplied with the dollar (1* 0.923702). After conversion, the euro amount is .92.

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; Essbase Analytic Services generates the data file during a data load and stores it on a disk.

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

data load rules. A set of criteria or rules that Analytic Services uses to determine how to load data from a text-based file, a spreadsheet, or a relational data set into an Analytic Services database.

data point. *See* cell.

data source. External data, such as a text file, spreadsheet file, relational database, or data warehouse that will be loaded into an Analytic Services database.

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 (DBA). An individual who administers database servers, such as Essbase Analytic Server, 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).

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.

descendant. Any member below a parent in the database outline. For example, in a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

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 dimension in an Essbase Integration Services metaoutline.

dimension build rules. In Essbase Analytic Services, 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 dimension in Essbase Analytic Services. 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 accounts, time, and standard (default) dimensions in Essbase Analytic Services.

drill-through report. Direct access by Spreadsheet Add-in users to data stored in the relational data source. Defined in Essbase Integration Services, a drill-through report is based on intersection levels (member sets) that Spreadsheet Add-in users double-click to view detail information that is not stored in the Essbase Analytic Services database.

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.

Dynamic Time Series. A process that is used to perform dynamic period-to-date reporting for all values associated with a query.

encoding. A method for mapping bit combinations to text characters for creating, storing, and displaying character text. Each encoding has a name; for example, UTF-8. Within a specific encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. *See also* code page, UTF-8 (Unicode Transformation Format, 8-bit encoding format), and locale.

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 an Essbase Analytic Services outline from the data source. When the Analytic Services outline is complete, Integration Server extracts data from the data source, performs the operations specified in the metaoutline, and loads the data into the Analytic Services database.

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

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

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

Essbase OLAP Server. *See* Analytic Server.

Essbase outline. *See* outline.

essbase.cfg. The name of an optional configuration file for Essbase Analytic Services. Administrators may enter parameters and values in this file to customize Analytic Server functionality. Some of the configuration settings may also be used with Analytic Services clients to override the Analytic Server settings.

ESSLANG. The Analytic Services environment variable that defines the encoding that Analytic Server uses to interpret text characters. *See also* encoding.

fact table. 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 Essbase Analytic Services, a value or item in a data source file that is retrieved from an Analytic Services 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. A method for controlling access to values and metadata in Essbase Analytic Services databases. A filter enables the placing of access restictions on specific data and metadata for specific users.

filter. (1) In Essbase Analytic Services, 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 Essbase Analytic Services, 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 an Analytic Services database. @VAR(Actual, Budget) is an example of a formula.

generation. A layer in a hierarchical tree structure that defines member relationships in a database. For example, generations are ordered incrementally from the top member of the dimension (generation 1) down to the child 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.

Hybrid Analysis. The integration of a relational database with an Essbase Analytic Services multidimensional database so that lower-level data remains in the relational database and is mapped to summary-level data residing in the Analytic Services database. Hybrid Analysis enables Analytic Services to take advantage of the mass scalability of a relational database while maintaining a multidimensional view of the data in a live environment.

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

index. (1) In Essbase Analytic Services, 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 Essbase Analytic Services, a buffer in memory that holds index pages.

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

index page. In Essbase Analytic Services, 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, an Analytic Services 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. Creates a relationship between two tables based on matching key column values in relational databases and as an SQL command.

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 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 Essbase Analytic Services 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.

locale. A computer setting that identifies the local language and cultural conventions such as the formatting of currency and dates, sort order of the data, and the character set encoding to be used on the computer. Analytic Services uses only the encoding portion of the locale. *See also* encoding and ESSLANG.

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 Essbase Analytic Services outline.

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

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

member combination. In Essbase Analytic Services, 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 an Essbase Analytic Services outline. The measures dimension contains actual members that are also member levels.

member load. In Essbase Integration Services, the process of adding new dimensions and members (without data) to an Analytic Services outline. *Contrast with* data load.

metadata. A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. 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 an Essbase Analytic Services outline from an OLAP model.

missing data (#MISSING). A marker indicating that data in the labeled location does not exist, contains no 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 organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection of a point for a set of dimensions.

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 is pulled from the relational data source. Essbase Integration Server accesses the OLAP Metadata Catalog to generate the SQL statements and the information required to generate an Essbase Analytic Services database outline.

OLAP model. In Essbase Integration Services, a logical model (star schema) that is created from tables and columns in a relational database. The OLAP model is then used to generate the structure of a multidimensional database.

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

online analytical processing (OLAP). A multidimensional, multiple-user, client-server computing environment for users who need to analyze consolidated enterprise data. OLAP systems feature functionality such as 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 Essbase Analytic Services. For example, an outline contains definitions of dimensions, members, and formulas.

parent. A member that has an aggregated branch of children 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 Server 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 Server parameter or Essbase Analytic Server configuration setting controls the duration and size of the queries made to the data source.

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

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 relational source 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 an Essbase Analytic Services 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). A database management system for accessing data in a relational database and storing data in the form of related tables. A RDBMS takes SQL statements entered by a user or contained in an application program and creates, updates, or provides access to the relational database.

restructure. In Essbase Analytic Services, 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* Analytic Server and Essbase Integration Server.

shared member. A member that shares storage space with another member of the same name. A storage property designates members as shared. The use of shared members prevents duplicate calculation of members that occur more than once in an Essbase Analytic Services outline.

sibling. A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are both children of the member, East, 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 Integration Server for drill-through reports. ODBC SQL, Native SQL, and stored procedures can be used when editing the standard SQL. The edited, user-defined SQL can be selected to improve data load performance when loading data into an Essbase Analytic Services 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. A logical model that represents your relational data in a form that mirrors that of OLAP data. A star schema contains a fact table and one or more dimension tables.

storage manager. *See* 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.

synonym. An alternate name for an object, such as a table or a view, in a relational database management system (RDBMS). Some RDBMSs use the term “alias” to refer to a synonym. Not all RDBMSs support synonyms.

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

table. In relational databases, a form of data storage in which data is stored in records comprised of fields. Each record is defined by a unique, or primary, key.

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. A 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 so that they return lists of members that have the specified UDA associated with them. UDAs can be applied to dense as well as sparse dimensions.

unary operator. A group of mathematical indicators (+, -, *, /, %) that define how roll-ups take place on the database outline.

Unicode. An approach to encoding character text such that thousands of text characters from hundreds of different languages are all supported within a single encoding form. *See also* encoding.

Unicode-mode Analytic Server. An Analytic Server with the setting enabled so that Unicode-mode applications can be created and non-Unicode-mode applications can be migrated to be Unicode-mode applications. *See also* Unicode-mode application.

Unicode-mode application. An Analytic Services application wherein character text is encoded in UTF-8, enabling users with their computers set up for different languages to share the application data.

union. An SQL command that is 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.

user-defined tables. Logical tables that you create in Essbase Integration Services, rather than in the RDBMS. These virtual tables, which behave as standard RDBMS views, can be used anywhere regular RDBMS tables or views are used. User-defined tables enable you to create and to edit OLAP models without altering your relational schema or modifying the SQL generated by Integration Services.

UTF-8 (Unicode Transformation Format, 8-bit encoding format). A Unicode encoding format that maps characters in a sequence of one to four bytes for each character. The first 128 characters of this encoding form use the same bit combinations as the common Latin1 encoding used for English characters. *See also* Unicode and encoding.

validation. (1) In Essbase Analytic Services, 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. A view can contain metadata and formatting information used to query an OLAP data source.

virtual tables. *See* user-defined tables.

virtual views. *See* user-defined tables.

write-back. The ability for a retrieval client, such as a spreadsheet, to update a database value.

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.

Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z